REDS:

Телекоммуникационные устройства и системы

№4

2023

СОДЕРЖАНИЕ

ВИЗУАЛИЗАЦИЯ ВОЗМУЩЕНИЯ ОПТИЧЕСКОЙ ПЛОТНОСТИ СРЕДЫ С ПОМОЩЬЮ ДАТЧИКА ВОЛНОВОГО ФРОНТА НА ЭФФЕКТЕ ТАЛЬБОТА

Казанцев Сергей Юрьевич
МТУСИ, профессор кафедры НТС, д.ф.-м.н., Москва, Россия
s.i.kazantsev@mtuci.ru

Пчелкина Наталия Владимировна
МТУСИ, доцент кафедры НТС, к.т.н., Москва, Россия
n.v.pchelkina@mtuci.ru

Смольский Алексей Александрович
МТУСИ, магистрант МТУСИ, Москва, Россия
a.a.smolskiy@mtuci.ru

Аннотация
Создан датчик волнового фронта на основе эффекта Тальбота. С целью использования этой установки в учебном процессе, были исследованы различные типы периодических решеток, отработаны методики их изготовления и возможности их применений в датчике волнового фронта. Продемонстрирована возможность визуализации возмущений оптической плотности на трассе лазерного пучка, что позволяет использовать датчик волнового фронта на основе эффекта Тальбота для создания адаптивных лазерных систем связи в свободном пространстве.

Ключевые слова: *эффект Тальбота, оптическая беспроводная связь в свободном пространстве, датчик волнового фронта, адаптивная оптика*

Введение

Известно, что при распространении лазерного излучения в атмосфере актуальной задачей является коррекция волнового фронта (ВФ) лазерного пучка с учетом возмущений оптической плотности, возникающих на трассе распространения лазерного излучения [1, 2].

Для того чтобы компенсировать искажения ВФ лазерного излучения применяются методы адаптивной оптики [2], где ключевым элементом является датчик волнового фронта (ДВФ), с помощью которого и производится регистрация ВФ [1]. При передаче данных в беспроводных оптических системах связи на дистанциях 3 км и выше, чтобы минимизировать геометрические потери, связанные с расходимостью лазерного излучения, применяются лазерные пучки с диаметром более 40 мм (для таких пучков в специальной литературе используют термин – широкоапертурные [3]). Применение серийных датчиков волнового фронта Шэка-Гартмана для регистрация волновых фронтов широкоапертурных лазерных пучков в ИК области спектра вызывает определенные проблемы [4, 5].

Однако кроме хорошо известных ДВФ Шека-Гартмана существует и другие ДВФ, в частности, для анализа ВФ широкоапертурных лазерных пучков ИК диапазона значительные преимущества имеет подход, однованный на эффекте Тальбота [3, 6]. Датчик волнового фронта на основе эффекта Тальбота позволяет визуализировать и исследовать оптические искажения, возникающие на трассе лазерного пучка с большими поперечными размерами. Отсутствие оптики позволяет применять данные датчики в рентгеновском и ИК спектральных диапазонах, а высокая чувствительность позволяет их использовать для контроля и юстировки различных оптических систем [3]. В прошлом, проблемы достижения высокого пространственного разрешения для изображений в ИК или рентгеновской области, а также необходимость их быстрой обработки, не позволили в полной мере проявить все преимущества ДВФ на основе эффекта Тальбота.

Прогресс в области создания цифровых оптических камер, разработка эффективных экранов для визуализации, а также совершенствование средств обработки изображений позволяет сегодня разрешить эти проблемы. Поэтому в последнее время проявляется значительный интерес к различным приложениям эффекта Тальбота [5-7].

Основой целью работы являлось разработка простого и удобного ДВФ на основе эффекта Тальбота, который можно использовать для анализа возмущений оптической плотности в атмосфере. С целью использования этой установки в учебном процессе [8], а также демонстрациях, иллюстрирующих эффект Тальбота, одной из задач работы являлось максимальное упрощение экспериментальной установки и увеличение ее надежности.

1. Теория датчика волнового фронта на основе эффекта Тальбота

В эффекте Тальбота после прохождения излучения через периодическую решетку и интерференции световых волн в ближней зоне Френеля воспроизводится распределение интенсивности на решетке, что происходит на кратных расстояниях, которые для квадратной решетки с периодом p, и, например, лазерного излучения с длиной волны λ, описываются формулой [3]:

$$L_n = \frac{2 \cdot p^2}{\lambda} \cdot nL_n = \frac{2 \cdot p^2}{\lambda} \cdot n, \quad (1)$$

здесь, p – период решетки, при этом необходимо чтобы выполнялось условие $p >> \lambda$; λ - длина волны монохроматического излучения, L_n – расстояние от решетки до плоскости воспроизведения, n – натуральное число (n = 1, 2, 3 …). Расстояние L_1 – называют расстоянием и длиной Тальбота [3]. Фактически, в эффекте Тальбота плоские когерентные волны источников лазерного излучения разлагаются на двумерной периодической решетке, при этом отверстия в решетке могут быть произвольными по форме, важна лишь периодичность их расположения.

Несложно показать [8], что эффект самовоспроизведения волнового фронта с периодической пространственной модуляцией полностью описывается теорией дифракции Френеля. Распределение интенсивности света в плоскостях перпендикулярных направлению распространения может быть вычислено на основе анализа зависимости комплексной амплитуды электрического поля E(x,y,z) вдоль распространения пространственно-модулированного пучка в зависимости от поля E_s(x,y) в плоскости источника. Очевидно, что периодическая решетка, через которую проходит свет, задает пространственную модуляцию светового пучка, тогда плоскость решетки, через которую прошел пучок, будет являться источником пространственно-модулированного светового поля. Выбрав ось z в направлении распространения пучка, и поместив начало координат в плоскость источника, который лежит в плоскости z=0, с использованием приближения Френеля можем записать:

$$E(x,y,z) = \sqrt{\frac{i \cdot k}{2\pi \cdot z}} \cdot \int_{-\infty}^{+\infty} E_s(x',y') \cdot exp\left(-\frac{i \cdot k}{2 \cdot z}[(x-x')^2 + \quad +(y-y')^2]\right) dx'dy', \quad (2)$$

где i – мнимая единица, $k = \frac{2\pi}{\lambda} k = \frac{2\pi}{\lambda}$ – волновой вектор. Для одномерной решетки этот интеграл еще больше упрощается:

$$E(x,z) = \sqrt{\frac{i \cdot k}{2\pi \cdot z}} \cdot \int_{-\infty}^{+\infty} E_s(x') \cdot exp\left[-\frac{i \cdot k}{2 \cdot z}(x-x')^2\right] dx'. \quad (3)$$

В случае когерентного источника с периодической модуляцией в плоскости z=0, амплитуда E_s(x) может быть представлена своим рядом Фурье:

$$E_s(x) = \sum_{-\infty}^{+\infty} E_n \cdot exp(i \cdot k_n \cdot x) E_s(x) = \sum_{-\infty}^{+\infty} E_n \cdot exp(i \cdot k_n \cdot x), \quad (4)$$

$$k_n = \frac{2\pi}{p} n k_n = \frac{2\pi}{p} n,\ n = 1, 2, 3, \quad (5)$$

здесь, p – период пространственной модуляции поля источника (период решетки), k_n – проекция волнового вектора плоской волны на ось x. Фактически в (4) записано разложение амплитуды поля источника на плоские волны. После подстановки разложения поля источника на плоские волны (4) в выражение (3) получим:

$$E(x,z) = \sqrt{\frac{i\cdot k}{2\pi\cdot z}} \sum_{-\infty}^{+\infty} E_n \cdot exp\left[ik_n \cdot \left(x + \frac{k_n}{2k} z\right)\right] \cdot \int_{-\infty}^{+\infty} exp\left[-\frac{ik}{2z}\left(x' - \left(x + \frac{k_n}{k} z\right)\right)^2\right] dx'. \quad (6)$$

Упрощая выражение (6) для комплексной амплитуды поля, получим:

$$E(x,z) = \sum_{-\infty}^{+\infty} E_n \cdot exp[i \cdot \phi_n] \cdot exp[i \cdot k_n \cdot x], \quad (7)$$

$$\phi_n = \frac{k_n^2}{2k} z \phi_n = \frac{k_n^2}{2k} z, \quad (8)$$

здесь $\phi_n \phi_n$ – набег фазы, который возникает у n-й гармоники плоской волны на расстоянии z от решетки. Из уравнения (7) следует, что амплитуда поля в любой плоскости $z=const$, также как и в плоскости источника представляется в виде ряда Фурье, и также будет иметь периодическую в направлении оси x структуру. Плоскость $z=const>0$, называется плоскостью изображения, а плоскость $z=0$ является плоскостью источника. Нетрудно теперь найти расстояния $z_t=L_n$, при которых распределение амплитуды электрического поля в плоскости изображения будет совпадать с распределением поля в плоскости источника, т.е. $E(x,z_t)=E(x,0)$. Для этого необходимо, чтобы набег фазы для всех плоских волны разложения (7) был кратным 2π, т.е. $\phi_n = 2\pi \cdot m$, где m – целое число. Если для первой гармоники набег равен 2π, то для n-й гармоники, используя (8) получим:

$$\phi_n = 2\pi n^2 \phi_n = 2\pi n^2. \quad (9)$$

Принимая в расчет (8) и приведенные выше соотношения для k и k_n, найдем расстояние Тальбота – минимальное расстояние, на котором распределение поля в плоскости изображений совпадает с распределением поля в плоскости источника:

$$z_t = \frac{2\cdot p^2}{\lambda} z_t = \frac{2\cdot p^2}{\lambda}. \quad (10)$$

Как видно это выражение совпадает с (1). Интенсивность светового поля в плоскости изображений (источника) будет пропорциональна квадрату амплитуды электрического поля и, естественно, воспроизводит такую же периодическую структуру.

В датчике ВФ на основе эффекта Тальбота смещение интерференционных максимумов в первой плоскости воспроизведения относительно положения, получаемого при падении идеального плоского фронта на периодическую решетку используется для определения локальных наклонов ВФ.

Таким образом, локальные наклоны ВФ, $\Delta\gamma$, измеряются в плоскостях воспроизведения. В приближении оптического клина:

$$\gamma = \frac{\Delta r}{L_1}, \quad (11)$$

$$\Delta\gamma = \frac{\sqrt{\Delta x^2 + \Delta y^2}}{L_1}, \quad (12)$$

определяется в параболическом приближении по изменению периодов Δp в плоскости воспроизведения:

$$R = \frac{p \cdot L}{\Delta p} R = \frac{p \cdot L}{\Delta p}, \quad (13)$$

здесь Δp – изменение периода в плоскости воспроизведения. Заметим, если ВФ плоский или сферический, то распределение интенсивности в плоскостях воспроизведения имеет периодическую структуру, которая подобна периодической структуре решетки.

2. Экспериментальные исследования датчика волнового фронта на основе эффекта Тальбота

Создание экспериментального стенда ДВФ на основе эффекта Тальбота осуществлялось аналогично [5, 8]. Схема экспериментальной установки приведена на рисунке 1. Подробное описание элементов стенда и методики регистрации тальбограмм содержится в [5, 8].

На рисунке 1 схематично показаны: ЛП – лазерный пучок; К – коллиматор (телескоп), который в [5] был образован двумя собирающими линзами $Л_1$ и $Л_2$; ОВ – оптическое возмущение, которое могло создаваться струей газа из баллончика или газовой горелкой; ПР – периодическая решетка; Ф – цифровой фотоаппарат; Э – экран для регистрации тальбограмм.

В настоящих экспериментах между периодической решеткой и экраном могла устанавливаться увеличивающая оптическая система (на схеме не показано), которая в простейшем случае может являться обычной линзой, что позволяет проецировать на экран увеличенное изображение распределения интенсивности в одной из фокальных плоскостей этой линзы. Данное решение особенно удобно при учебных демонстрациях эффекта Тальбота, когда его надо показывать большой аудитории.

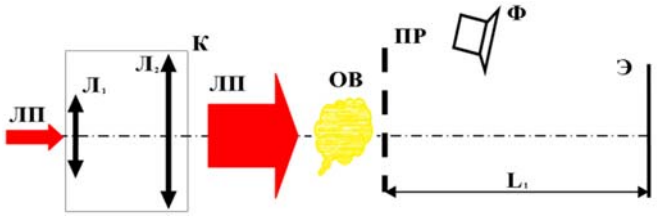


Рис. 1. Схема экспериментальной установки для исследования эффекта Тальбота и ДВФ на его основе

С учетом специфики применения этой установки в экспериментах по анализу оптических возмущений на воздушных трассах и необходимости сделать ее максимально мобильной, было внесено несколько модификаций с целью максимального упрощения ДВФ и сокращения времени его настройки (юстировки). В частности, по сравнению с той, что описана в [5,8], вместо газовых и твердотельных лазеров был использован лазерный диод, излучающей на длине волне 660 нм. Лазерный диод через оптическое одномодовое волокно подключался к коллиматору C80FC-B фирмы THORLABS, предназначенного для работы в ближнем ИК-спектре. Этот коллиматор позволял использовать, как видимые лазеры, так и ИК лазерный диод на 850 нм. Лазер на длинах волн вблизи 850 нм используется в системах спутникового квантового распределения ключей [9,10], поэтому в дальнейшем представляет исследовать искажение ВФ, вызванных возмущениями оптической плотности атмосферы, именно на этой длине волны.

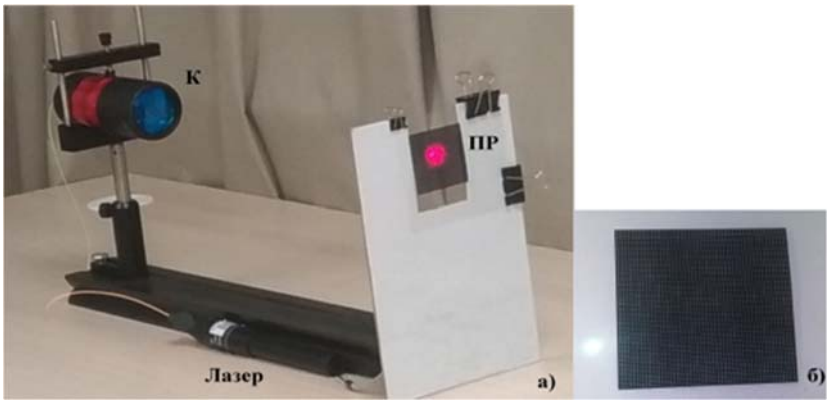


Рис. 2. Фотографии: а) – внешний вид экспериментальной установки в процессе юстировки; б) – фотография периодической решетки, напечатанной на пленке

На рисунке 2а показан внешний вид экспериментальной установки. На фотографии помечены: коллиматор; лазер; периодическая решетка. В качестве периодической решетки применялась лавсановая пленка толщиной 60 мкм, на которой была с помощью лазерного принтера напечатана квадратная решетка [11].

В зависимости от величины оптических возмущений на трассе лазерного пучка использовались ПР с разным периодом: от 0.4 мм до 1.3 мм, необходимость изменения периода решетки определяется возможностью четкого определения смещения интерференционного максимума, поскольку при смещении этих максимумов более чем на период решетки не позволяет их различить, а кроме того приближенные формулы (11) и (13) уже нельзя использовать.

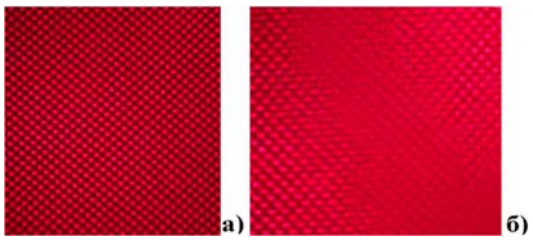


Рис. 3. Фотографии тальбограмм:
а) – лазерный пучок не возмущался; б) – пучок прошел через оптическое возмущение

На рисунке 3 показаны тальбограммы полученные для невозмущенного лазерного пучка (рис.3а) и для лазерного пучка прошедшего через восходящий тепловой поток от газовой горелки. Лазер с длиной волны 660 нм и средней мощностью 20 мВт, подключенный к коллиматору (см. рис.2а) позволял проводить настройку системы без использования специальной ИК камеры. На рис. 3б по смещению святящихся точек, а также и изменению периода между ними, может быть восстановлено изменение показателя преломления во всей области пространства, через которую пропускается широкий лазерный пучок, вышедший из коллиматора. Наши исследования также показали, что полезным дополнительным элементом установки, является увеличительная система в виде обычной линзы, закрепленной на фиксированном расстоянии от непрозрачного экрана, что позволяет проецировать на экран увеличенное в несколько раз изображение тальбограммы в первой плоскости воспроизведения.

Заключение

В статье описаны теоретические и технические принципы построения надежного и простого в настройке ДВФ на основе эффекта Тальбота, который может быть использован для анализа оптических возмущений в атмосфере. Показано, что созданный нами экспериментальный стенд является удобным и высокочувствительным ДВФ широкоапертурных лазерных пучков видимого и ИК диапазона.

Созданная нами установка может быть использована для отработки математических алгоритмов анализа изображений интерферограмм, проецируемых на экране, с целью построения поверхности ВФ

и получения количественной информации о величине оптических возмущений среды на трассе лазерного пучка, что является необходимым для создания адаптивных оптических систем [1]. Созданный ДВФ может также использоваться для контроля параметров широкоапертурной оптики применяемы в космических системах связи с квантовым распределением ключей, описанных в [10] и контроля потерь мощности в системах атмосферной квантовой связи [11].

Литература

1. *Больбасова Л.А., Лукин В.П.* Адаптивная оптика: учебное пособие. 2021.

2. *Больбасова Л.А., Лукин В.П.* Исследования атмосферы для задач адаптивной оптики // Оптика атмосферы и океана. 2021. Т. 34. № 4(387). С. 254-271. DOI 10.15372/AOO20210403.

3. *Коряковский А.С., Марченко В.М., Прохоров А.М.* Дифракционная теория метода Тальбот-интерферометрии и диагностики широкоапертурных волновых фронтов // Труды ИОФАН. 1987. Т. 7. С. 33-92.

4. *Игнатьев А.Б.* и др. О возможности контроля волнового фронта широкоапертурного HF (DF)-лазера методом тальбот-интерферометрии // Квантовая электроника. 2008. Т. 38. № 1. С. 69-72. EDN TTEWRX.

5. *Волкова Л.В.* и др. Датчик волнового фронта широкоапертурных лазерных пучков и его применения // Журнал технической физики. 2022. Т. 92. № 9. С. 1410-1414. DOI 10.21883/JTF.2022.09.52933.49-22. EDN CMIHSM.

6. *Podanchuk D.V.* et al. Adaptive wavefront sensor based on the Talbot phenomenon // Applied optics. 2016. Vol. 55. No. 12. P. B150-B157.

7. *Srisuphaphon S., Buathong S., Deachapunya S.* Realization of an optical vortex from light-emitting diode source by a vortex half-wave retarder and using Talbot effect based detection // Optics & Laser Technology. 2022. Vol. 148. P. 107746.

8. Оптика: Лабораторный практикум. Том Часть 3. Москва: Московский Политех, 2022. 72 с. EDN ORXTHB.

9. *Liao S.K.* et al. Satellite-to-ground quantum key distribution // Nature. 2017. Vol. 549. No. 7670, pp. 43-47. DOI: 10.1038/nature23655

10. *Sidhu J.S.* et al. Advances in space quantum communications // IET Quantum Communication. 2021. Vol. 2. No. 4, pp. 182-217. DOI: 10.1049/qtc2.12015

11. *Bolotov D.V.* et al. A Method for Estimating Losses in a Quantum Channel for Implementing Quantum Key Distribution Technology for Atmospheric Laser Communication Terminals // Wave Electronics and Its Application in Information and Telecommunication Systems. 2022. Vol. 5, No. 1, pp. 57-61. EDN WFQNYF.

ЦИФРОВАЯ ГИГИЕНА: КОНТРОЛЬ ДОВЕРИТЕЛЬНЫХ ОТНОШЕНИЙ В ИНТЕРНЕТЕ

Михалевич Игорь Феодосьевич
Российский университет транспорта (МИИТ), доцент, к.т.н., с.н.с., Москва, Россия
mif-orel@mail.ru

Аннотация
Статья продолжает цикл публикаций о совокупности правил поведения в информационном обществе, объединенных термином «цифровая гигиена». Отмечается, что их несоблюдение влечет риски материального ущерба и морального вреда для всех без исключения участников информационного взаимодействия. В данном случае эти правила рассмотрены в отношении установления и контроля доверительных отношений в интернете, приведены примеры их применения.

Ключевые слова: *Доверенная третья сторона, информационная безопасность, информационное взаимодействие, рейтинг безопасности сайта, ущерб, цифровая гигиена*

Введение

Следуя [1], цифровую гигиену можно рассматривать как совокупность правил для физических лиц, соблюдение которых защищает от материального ущерба и/или морального вреда их самих, их семьи, родных и близких, а также связанные с ними коллективы и организации (далее – пользователи) от угроз, которые в настоящее время связаны, в первую очередь, с глобальной сетью Интернет. И если выполнение рекомендованных правил не защитит от интернет-угроз полностью, то, как минимум, снизит возможность возникновения такого ущерба или понизит его уровень.

В [1,2] основное внимание было уделено правилам, основанным на обеспечении надлежащего «цифро-гигиеничного» состояния оконечных устройств (персональных компьютеров, смартфонов, планшетов, …). В этой статье рассмотрен вопрос, насколько обоснованы ожидания пользователей в отношении безопасности ресурсов, с которыми они взаимодействуют в интернете, и как можно повысить безопасность такого взаимодействия, не обладая специальной подготовкой «борца» с киберпреступностью.

Какие отношения в интернете можно считать доверительными?

Чтобы сформулировать, какие отношения являются доверительными, рассмотрим отношения сторон при совершении юридически значимых действий. Таковыми считаются действия, влекущие правовые последствия в виде возникновения прав или обязанностей, имеющих значение для суда.

Например, физическое лицо хочет открыть счет в банке и внести на него денежные средства.

В этом случае возникает две стороны отношений в лице вносителя средств и получателя средств в лице банка соответственно. Когда отношения между ними можно будет считать доверительными? Когда каждый из них предоставит для начала доказательства, что является тем, за кого себя выдает, и далее убедит вторую сторону на наличие права совершения планируемых юридически значимых действий.

Какие доказательства может предоставить вноситель? Это может быть подписанный им листок, на котором будут указаны фамилия, имя, отчество, место и дата рождения, место регистрации и место фактического проживания и т.д. Вызовет ли такой документ о вносителе доверие у банка? Каждый имеет возможность убедиться в том, что такой самоподписанный документ доверия у банка не вызовет. Сразу оговоримся, если это действительно банк, а не тот, кто его имитирует и готов принять деньги у кого угодно, в том числе для того, чтобы тут же с ними скрыться.

Вноситель может попытаться исправить ситуацию и представить аналогичный документ, подписанный, например, директором жилищной конторы и заверенный ее печатью. Но и это не поможет в случае с настоящим банком.

Можно и далее рассматривать аналогичные попытки вызвать к себе доверие банка. Но в действительности успех ждет только в случае представления документа о личности, удостоверенного третьей стороной, не доверять к которой банк не вправе. Таким документом, который содержит все вышеприведенные данные, является, например, паспорт гражданина РФ. Третьей стороной в таком случае (далее - доверенной третьей стороной), выступает МВД России в лице его территориального подразделения.

А какие доказательства может предоставить банк? Например, листок, на котором написаны название банка, место нахождения и т.д., даже заверенный печатью, на которой все это будет повторено, дает ли достаточные основания доверять банку и рисковать своими деньгами. Осмелюсь ответить – НЕТ!!! Однако в интернете такого рода истории, связанные с перечислениями денег, имеют место быть достаточно часто.

Как известно, надеюсь, каждому, настоящий банк не затруднит предоставить вносителю документы, которые легко проверить. В их число входит, как минимум, выписка из единого государственного реестра юридических лиц и документы о филиале или отделении банка, если таковые имеются, и лицензия Банка России. Они содержат достаточно сведений, подтвержденных доверенными третьими сторонами в лице инспекции Федеральной налоговой службы РФ и уполномоченного органа Банка России.

Такие аналогии можно продолжить в отношении купли-продажи имущества (квартир, домов, дач, транспортных средств, …) и множества иных юридически значимых действий, совершаемых с участием, чаще всего, незнакомых сторон.

Если живое общение в банке или у нотариуса может защитить добросовестную сторону от недобросовестных действий другой стороны, то в интернете такая возможность кратно сокращается. Более-менее эта возможность может обеспечиваться приложениями (банков, ритейлеров, …), которые прошли соответствующие проверки по требованиям безопасности информации, успешность которых подтверждена сертификатами уполномоченных органов, которыми в настоящее время в России являются ФСБ России и ФСТЭК России.

В остальных случаях безопасность своих активов, и не только денежных, должны обеспечивать сами пользователи.

Выше в перечислении ФСБ России указана первой не случайно, так как безопасность отношений в интернете обеспечивается средствами криптографической защиты информации, что входит в сферу регулирования ФСБ России.

Из изложенного можно заключить, что в интернете должна существовать некая система, создающая аналогичные отношения между взаимодействующими сторонами. И такая система действительно есть и обеспечивается она SSL-сертификатами (SSL – Secure Sockets Layer, т.е. – уровень защищённых сокетов). Эти сертификаты по факту являются электронными документами о стороне, в данном случае присутствующей интернете.

Эти сертификаты имеют разновидности, что позволяет отличить самоподписанный сертификат (наподобие листочка с подписью о том, что «я – это я»), от сертификата, выданного доверенной третьей стороной, наподобие МВД России, Федеральной налоговой службой России и т.п.

Сертификаты доверенной третьей стороны имеют, в свою очередь три градации (DV, OV, EV), в зависимости от того, какие сведения об участнике информационного взаимодействия в интернете они содержат [3]. При минимальном объеме проверок сертификат типа DV подтверждает соответствие доменного имени серверу и сайту. При максимальном объеме проверок сертификат типа EV будет содержать расширенный объем сведений об участнике информационного взаимодействия, включая юридические данные. Как будет показано далее категория сертификата может быть легко проверена и риски, возникающие в случае установления отношений с его владельцем, могут быть учтены.

На основе SSL-сертификатов могут быть получены, в частности, следующие сведения, обеспечиваемые функциями криптографических протоколов [1]:

– способ аутентификации источника, используемый для проверки принадлежности данных автору;

– уровень аутентификации сторон, позволяющий определить соответствие сведений о сторонах с теми, которые были предоставлены ими при сертификации;

– как осуществляется защита от повтора и чем обеспечивается невозможность использования одних и тех же данных более одного раза (например, более одного перечисления денег с использованием одного платежного документа);

– какой используется алгоритм шифрования данных для обеспечения их конфиденциальности;
– способ обеспечения целостности данных;
– способы обеспечения неотказуемости сторон и подотчетности совершаемых ими действий в интернете (направления писем, документов и т.п. и их получения).

Насколько можно доверять внешним признакам возможности или невозможности доверительных отношений?

Общепринятыми признаками, указывающими на возможность установления доверительных отношений с сайтом, является наличие в адресной строке страницы сайта записи следующего вида: https://www.xxx.qyx//, – часто дополняемой визуально «замочком», как это представлено на рисунке 1.

Однако есть риск в сроке браузера увидеть такую запись: http://www.xxx.qyx//, – и не обнаружить разницы с предыдущей, особенно, если отсутствует визуализация или явное предупреждение, варианты которого представлены на рисунке 2.

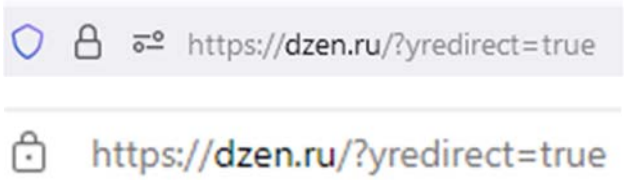


Рис. 1. Визуализация возможности установления доверенных отношений с сайтом с использованием браузеров Microsoft Edge (сверху) и Firefox (внизу)

Отметим, что как в первом случае (рис. 1), так и во втором (рис. 2), полученная только таким образом информация может оказаться неполной для принятия окончательного решения о возможности или невозможности доверительных отношений с сайтом.

Рис. 2. Визуализация предупреждения о нежелательности установления отношений с сайтом с использованием браузеров Microsoft Edge (сверху) и Firefox (внизу)

Как можно более полно проверить эти возможности или невозможности, не прибегая, пока, к специализированным средствам контроля, а используя только внутренние механизмы браузеров?

Сразу отметим, что эти механизмы во многом идентичны, но, к сожалению, не все браузеры характеризуются «благосклонным» и «доброжелательным» отношением к простым пользователям, желающим получить более полную информацию в целях контроля возможности установления доверительных отношений с сайтами.

Чтобы в этом убедиться достаточно «нажать» на «замочек» и пройти дальше для ознакомления с подробностями соединения, чтобы увидеть в одном случае, например, то, что представлено на рисунках 3 и 4 для закрытого «замочка».

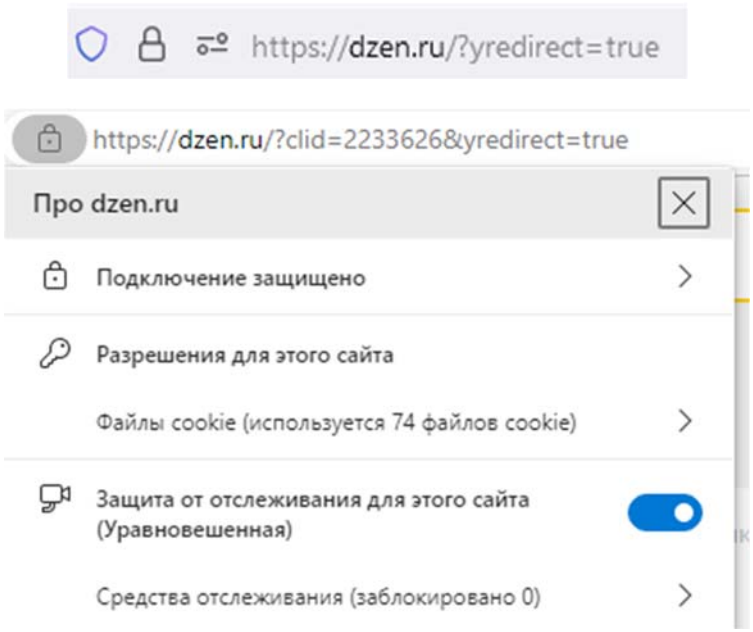
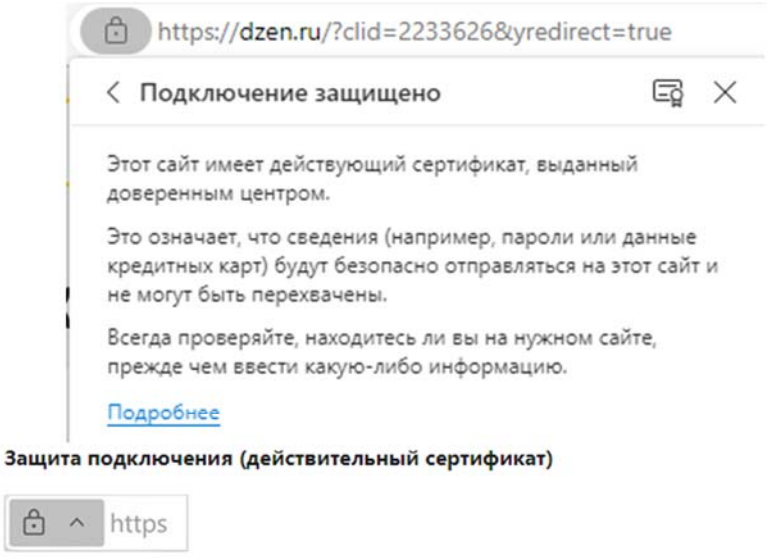


Рис. 3. Информация о защищенном сайте, предоставляемая браузером Microsoft Edge



Защита подключения (действительный сертификат)

- На веб-сайте, который вы посещаете, есть действительный сертификат, выданный доверенным доверенным органом. Информация, отданная на сайт и с нее, защищена и не может быть перехватиться злоумышленником. Однако даже веб-сайты с действительными сертификатами могут иметь неудовлетворительную репутацию, поэтому всегда проверяйте URL-адрес в адресной панели, чтобы убедиться, что вы на нужном сайте, прежде чем вводить какие-либо сведения.

Рис. 3 (продолжение). Информация о защищенном сайте, предоставляемая браузером Microsoft Edge

Из рисунка 3 видно, что конкретики в отношении характеристик для подтверждения доверительности предоставлено немного. Пользователь должен полностью довериться владельцу браузера кампании Microsoft.

Совершенно иная картина предстает на рисунке 4.

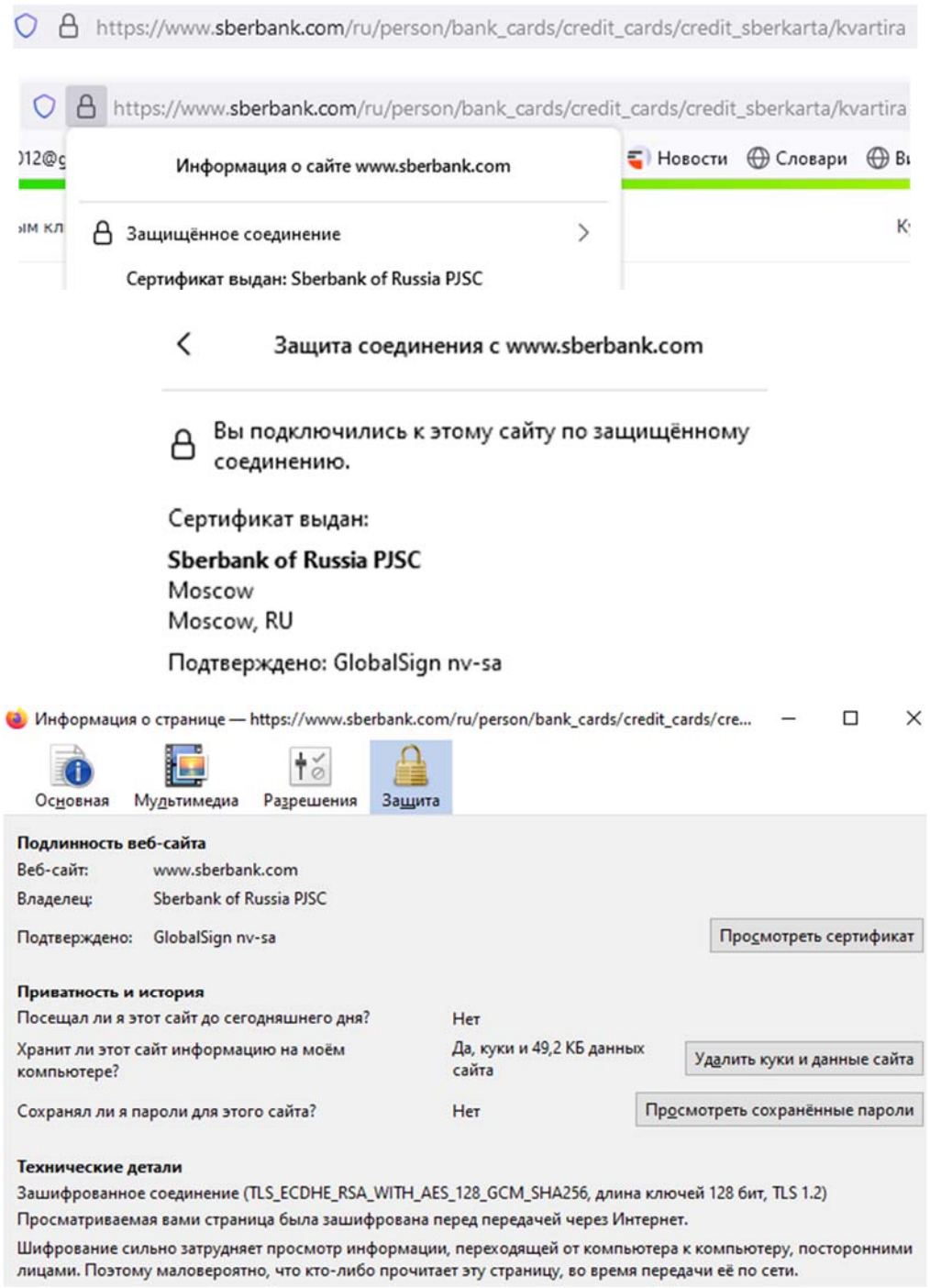


Рис. 4. Информация о защищенном сайте, предоставляемая браузером Firefox

Из рисунка 4 видно, что в Firefox браузере состояние страницы сайта с адресом: https://www.sberbank.com/ru/person/bank_cards/credit_cards/credit_sberkarta/kvartira , – из недоверенного изменилось на доверенное и указана причина – сертификат, который выдан удостоверяющим центром Sberbank of Russia PJSC, признан браузером доверенным.

Более того, информация о соединении сопровождается техническими деталями: TLS_ECDHE_RSA_WITH_AES_128_GSM_SHA256, длина ключей 128 бит, TLS 1.2, – отражающими спецификацию криптографического протокола, криптографические методы, характеристики ключевой информации и способы ее получения, которые будут использоваться для защиты сторон в случае продолжения информационного взаимодействия.

Аналогичную картину отношения к пользователям можно наблюдать в отношении сайтов, которые браузеры считают недоверенными (рис. 5 и 6).

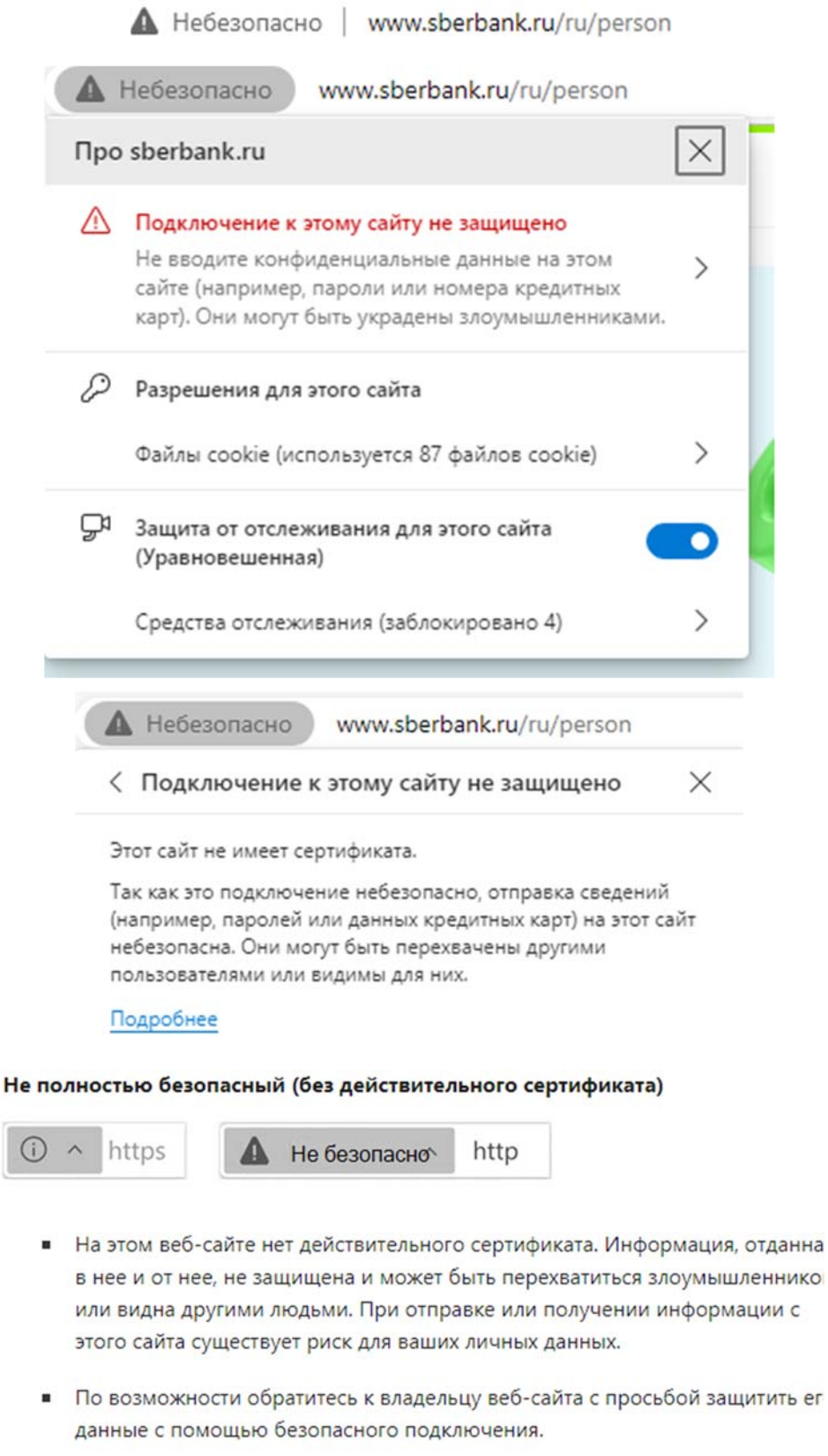


Рис. 5. Информация о недоверенном сайте, предоставляемая браузером Microsoft Edge

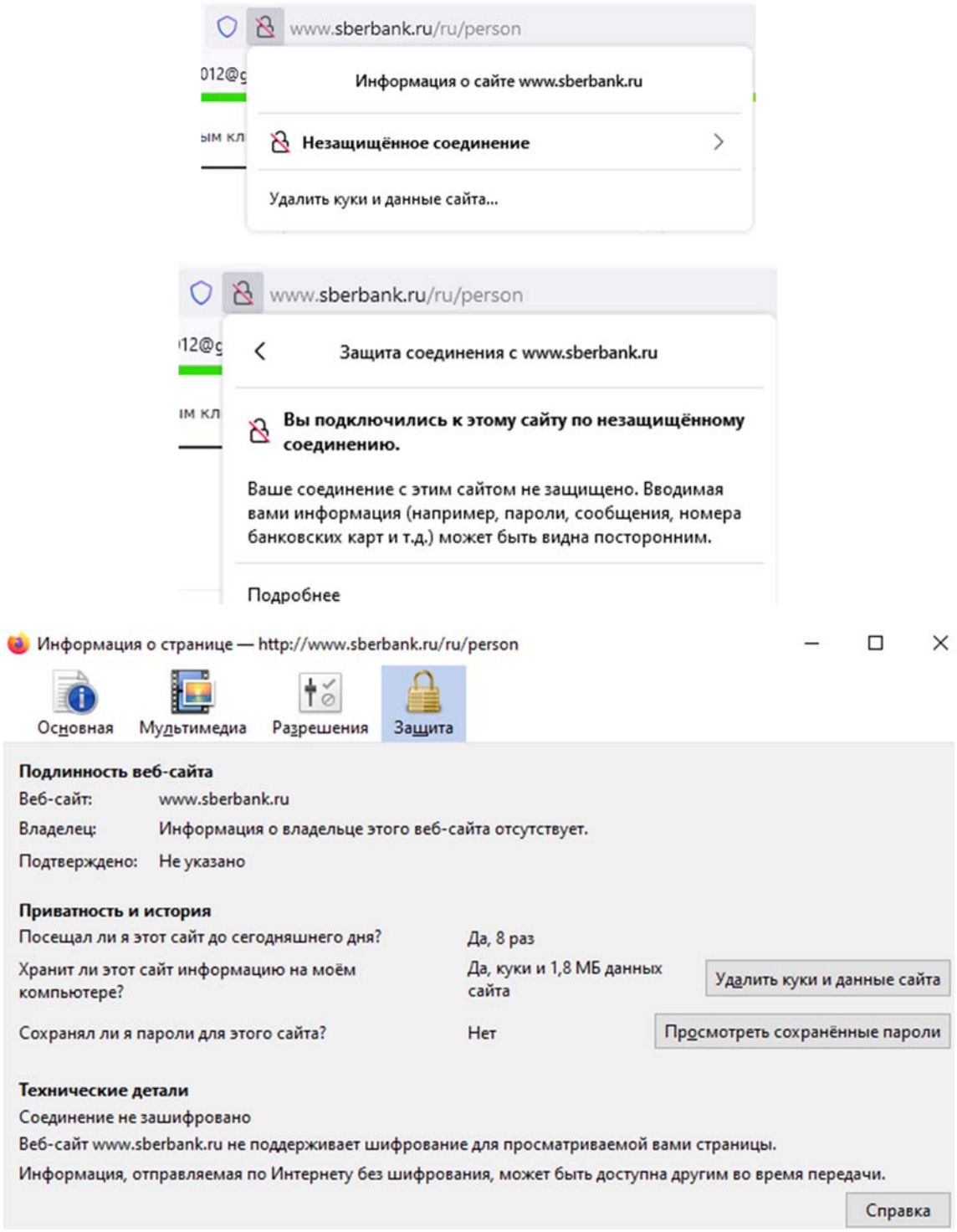


Рис. 6. Информация о недоверенном сайте, предоставляемая браузером Firefox

Есть ли простые и более полные способы проверки признаков возможности или невозможности доверительных отношений?

Представленное выше исследование показывает, что ориентироваться только на внешние признаки не стоит. Поэтому существует достаточно большое число платных и бесплатных инструментов, позволяющих более обоснованно прийти к тому или иному выводу, в том числе с использование количественных показателей.

Для количественной оценки введем в рассмотрение следующий мультипликативный показатель: уровень подтверждения доверенности сайта – УПДС. С учетом описанных выше функций криптографических протоколов, направленных на обеспечение доверительных отношений в интернете, запишем формулу (1) для его вычисления:

$$УПДС = ВПр \times ВКлС \times ВШ, \quad (1)$$

где, ВПр – версионность поддерживаемых криптографических протоколов (SSL, …, TLS 1.2, TLS 1.3); ВКлС – версионность системы управления криптографическими ключами, их генерацией и распространением; ВШ – версионность поддерживаемых шифров, отражающая стойкость шифров.

Все эти показатели могут быть вычислены при анализе ранее приведенной спецификации криптографического протокола:

TLS_ECDHE_RSA_WITH_AES_128_GSM_ _SHA256, длина ключей 128 бит, TLS 1.2, (2)

если ее представить в виде коэффициентов *k*- при соответствующих составляющих спецификацию методов, как показано, для примера, в (3):

$$УПДС = TLS1.2 \times k\text{-ECDHE \& RSA} \times k\text{-AES\_128\&GSM} \times k\text{-SHA256} \times k\text{-(длина ключей 128 бит)} \quad (3)$$

Значения УПДС, при которых может быть принято то или иное решение, должны устанавливаться индивидуально каждым пользователем, исходя из соображений о допустимом размере (величине) ущерба, который может наступить в результате ошибочного решения. Эта тема выходит за рамки настоящей статьи, поэтому далее не развивается. Но возможные примеры вариантов ее применения показаны ниже.

Для оценки доверия к сайтам имеется большое число специализированных инструментов, в том числе бесплатных. Топ-10 инструментов для проверки SSL, TLS и последних уязвимостей сайтов выгладит следующим образом [4-9]:

1. SSL Labs от Qualys
2. SSL Checker
3. Symantec
4. Wormly
5. DigiCert
6. SSL Server Security Test
7. SSL Analyzer
8. SSL Checker
9. HowsMySSL
10. SSL Checker

Длительное время рейтинг возглавляют инструменты от Qualys, что говорит о высоком доверии к ним со стороны специалистов по информационной безопасности. В то же время эти инструменты являются бесплатными. Они обладают простотой и высокой наглядностью результатов тестирования, понимание которых не требует специальных знаний, что позволяет рекомендовать их к использованию повсеместно в интересах цифровой гигиены.

Для проверки сайта достаточно обратиться по адресу https://www.ssllabs.com/ssltest/ и в окне «Hostname» ввести имя проверяемого сайта (рис. 7).

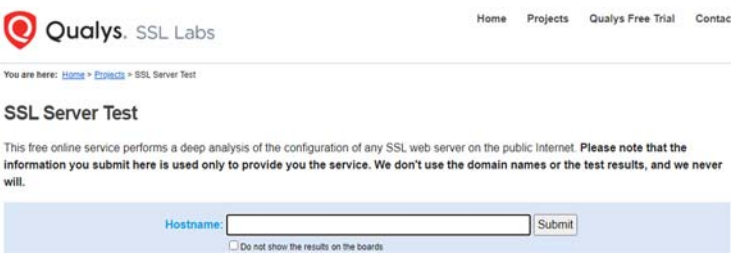


Рис. 7. Окно ввода имени проверяемого сайта

Спустя незначительное время инструмент предоставит отчет, содержащий оценку сайта и подробную информацию о том, на чем она основана.

Для приведенных на рис. 3 и 4 примеров результаты оценки сайтов представлены на рис. 8 и 9.

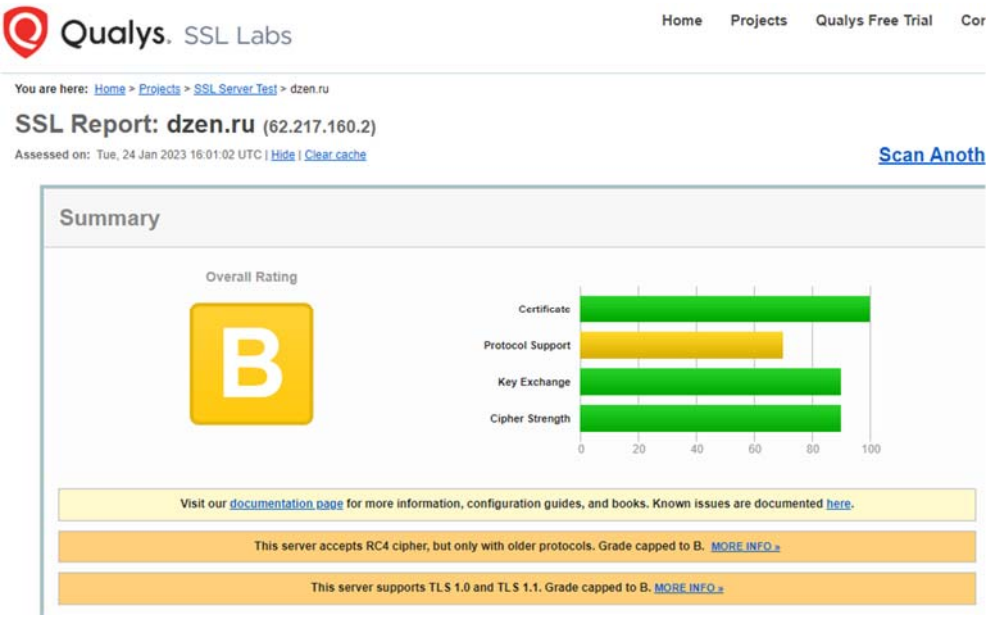


Рис. 8. Результаты инструментальной проверки сайта Яндекс Дзен

Средняя оценка (рейтинг) безопасности сайта уровня B означает, что значение интегрированного показателя равно или более 0,65 [10], если значения частных показателей измеряется в диапазоне от нуля до единицы.

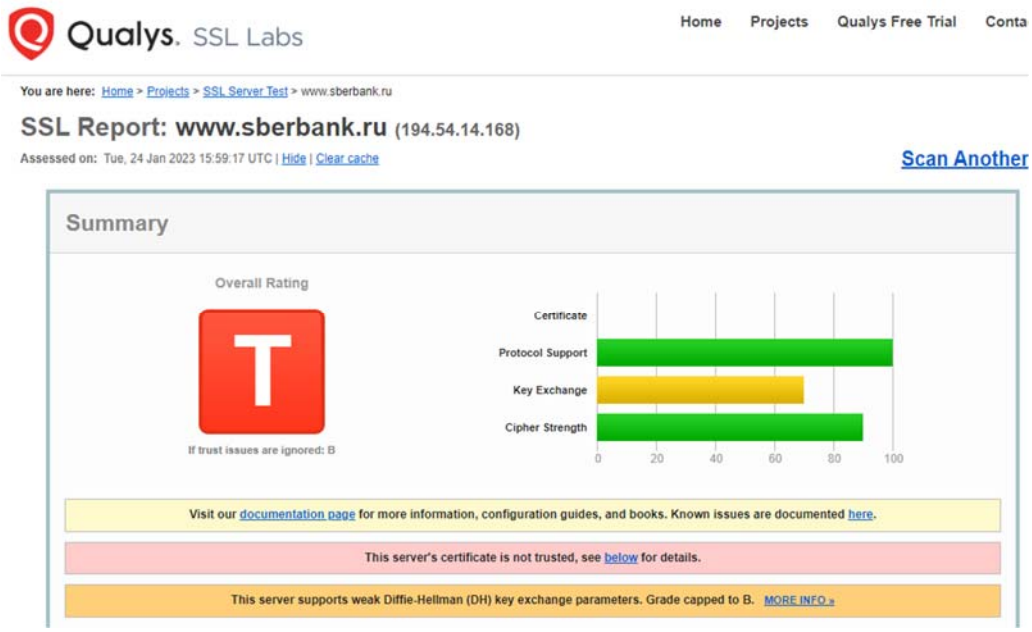


Рис. 9. Результаты инструментальной проверки сайта Сбербанк

Для данного сайта инструмент также указал оценку B, сопроводив ее комментарием, что эта оценка справедлива для случая, если вопросы доверия игнорируются. Причина такого комментария видна из рисунка 9, где в строке оценки сертификата информация не отображена. По какой причине это происходит с российскими сайтами было сказано ранее.

Заключение

Проведенное в работе исследование особенностей установления и контроля доверительных отношений в интернете основано на аналогии с действиями участников информационного общества вне интернета. Это позволяет, на взгляд автора, лучше понять сущность предлагаемых правил цифровой гигиены и их важность.

Рассмотрены варианты и приведены примеры контроля доверительности отношений в интернете с использованием внутренних механизмов браузеров и с применением специализированных средств, доступных на бесплатной основе.

Статья предназначена для широкого круга читателей. Вместе с этим ее материалы могут использоваться в учебном процессе по разным специальностям и направлениям подготовки, что подтверждает педагогический опыт автора. Ведь сегодня не найти ни одной учебной дисциплины, где обучающими и обучаемыми вовсе не применялся бы интернет.

Литература

1. *Михалевич И.Ф. Михалевич И.Ф.* Цифровая гигиена информационного общества: влияние пандемии COVID-19 // REDS: Телекоммуникационные устройства и системы, № 3-2022. С. 10-17.

2. *Михалевич И.Ф.* Цифровая трансформация систем управления в условиях пандемии COVID-19 // REDS: Телекоммуникационные устройства и системы/ № 4-2021. С. 26-32. (доступ 05.09.2021).

3. Что такое SSL-сертификат – определение и описание. https://www.kaspersky.ru/resource-center/definitions/what-is-a-ssl-certificate (доступ 19.03.2022).

4. Chandan Kumar, "10 Online Tools to Test SSL, TLS and Latest Vulnerability". https://geekflare.com/ssl-test-certificate/ (доступ 24.01.2023).

5. Blogger D.D. Top 10 Online Tools to Test SSL, TLS and Latest Vulnerability. https://www.dignitasdigital.com/blog/online-tools-to-test-ssl-tls-vulnerability/ (доступ 12.11.2022).

6. 10 Online Tool to Test SSL, TLS and Latest Vulnerability. https://luvunix.wordpress.com/2018/02/22/10-online-tool-to-test-ssl-tls-and-latest-vulnerability/ (доступ 05.09.2021).

7. Top 10 Online Tools to Test SSL, TLS and Latest Vulnerability. https://ctrlr.org/top-10-online-tools-to-test-ssl-tls-and-latest-vulnerability/ (доступ 15.02.2022).

8. 10 Free SSL / TLS Diagnostic Tools for the Webmaster. https://sudonull.com/post/11416-10-Free-SSL-TLS-Diagnostic-Tools-for-the-Webmaster (доступ 11.01.2023).

9. *Tim Keary*. 10 Best SSL Checkers. https://www.comparitech.com/net-admin/best-ssl-checkers/ (доступ 05.09.2021).

10. SSL Server Rating Guide. https://www.ssllabs.com/projects/rating-guide/index.html (доступ 05.09.2021).

ВЗАИМОСВЯЗЬ ПАРАМЕТРОВ ОЦЕНКИ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Шведов Андрей Вячеславович,
МТУСИ, старший преподаватель кафедры СИТиС, Москва, Россия
a.v.shvedov@mtuci.ru

Яковенко Наталья Викторовна,
МТУСИ, старший преподаватель кафедры СИТиС, Москва, Россия
nv1906.iakovenko@yandex.ru

Коровушкина Вероника Максимовна,
МТУСИ, магистрант гр. М092201(75), Москва, Россия
vkorovushkina10@gmail.com mailto:a.v.shvedov@mtuci.ru

Гадасин Денис Вадимович,
МТУСИ, доцент кафедры СИТиС, к.т.н., Москва, Россия
dengadiplom@mail.ru

Аннотация
Увеличение уровня сложности технологий провоцирует повышение уровня требований к надежности информационных систем. На данный момент нет унифицированного набора инструментов для оценки и повышения данного показателя. В работе рассматривается модификация жизненного цикла программного обеспечения в целях управления процессом оценки надежности, а также предложен способ корреляции метрик надежности, как решение проблемы комплексного подхода к оценке надежности информационных систем.

Ключевые слова: *надёжность, оценка надежности, метрики надежности, корреляция метрик надежности*

Введение

Согласно статистике, опубликованной группой компаний “Деловой профиль”, а также [1,2] в течение последних 10 лет объем IT рынка увеличивается ежегодно минимум на 10%. По большей степени это происходит за счет программного обеспечения. Такой скачок обуславливается тенденцией на автоматизацию повседневных и рутинных процессов. С увеличением выполняемых задач программное обеспечение стало более сложным, при этом его реализация представляет собой информационную систему с множеством узлов и модулей для выполнения каждой необходимой функции. Соответственно, чем сложнее информационная система, тем острее возникает вопрос о ее качестве и надежности.

Для того, чтобы управлять надежностью программного обеспечения необходимо уметь его оценивать. В [3,4,5] был предложен и рассмотрен подход выделения определенного набора метрик надежности и вычисления их численного значения на основе математических моделей.

К рассмотренным метрикам относятся:

- Средняя наработка на отказ;
- Среднее время восстановления;
- Вероятность нахождения в работоспособном состоянии;
- Интенсивность отказов.

Для каждой метрики была предложена математическая модель, с помощью которой можно определить ее численное значение. Например, вероятность нахождения в работоспособном состоянии можно определить по модели Коркорэна.

Особенностью данной модели является зависимость результата не от времени, а только от результата проводимых тестов и выявленных ошибок или отказов.

Показатель вероятности нахождения в работоспособном состоянии оценивается по формуле:

$$R = \frac{N_0}{N} + \sum_{j=1}^{K} \frac{Y_j (N_j - 1)}{N}, \quad (1)$$

где N_0 – количество корректных запусков систем; N – общее количество тестовых запусков; K – количество типов ошибок; Y_j – вероятность выявления при тестировании ошибки i-типа.

Но знание численных оценок характеристик надежности, как отдельных параметров, не может дать комплексное понимание, в какой мере надежна информационная система [6]. Поэтому, для формирования комплексной оценки надежности информационной системы необходимо учитывать корреляцию всех ее параметров.

Кроме выявления взаимосвязи между метриками необходимо создать условия для возможности управления данным процессом. Для этого нужно определить место оценки надежности в жизненном цикле программного обеспечения.

Место процесса оценки надежности в жизненном цикле программного обеспечения

Сейчас предложено множество видов жизненного цикла программного обеспечения, но все разновидности циклов подразумевают под собой несколько этапов: планирование, анализ требований, проектирование, написание и отладка программного кода, тестирование, внедрение и поддержка. Базовая схема процесса разработки программного обеспечения представлена на рисунке 1.

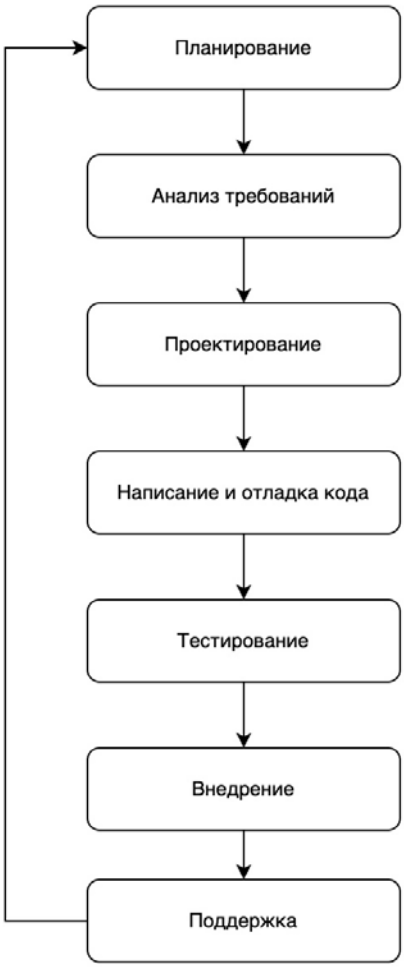


Рис. 1. Базовая схема процесса разработки ПО

На этапе тестирования выявляются те ошибки, которые были допущены на предыдущих этапах жизненного цикла программного обеспечения, за исключением тех ошибок, которые были изначально заложены в техническом задании. Также на этапе тестирования выполняется контроль качества информационной системы. Если программное обеспечение названо качественным, значит, оно надежно. Исходя из этого, надежность определяется потребителем исходя из того, как оно было протестировано.

Такой подход является некорректным, ведь надежность является одним из параметров качества, который необходимо рассчитывать отдельно. Оценивать надежность программного обеспечения только на основе количества выявленных ошибок нельзя в связи с некоторыми особенностями использования ПО конечными пользователями. К примеру, процесс тестирования занимает короткий промежуток времени относительно времени эксплуатации информационной системы или же на компьютерах пользователей может быть установлено другое программное обеспечение, которое не совместимо с данной информационной системой. Соответственно, для оценки надежности программного обеспечения необходимо выработать отдельный процесс и определить его место в жизненном цикле ПО.

Первоначально необходимо проанализировать требования к продукту для определения набора приоритетных метрик и моделей в зависимости от специфики тестируемого программного обеспечения. После этого можно приступать к сбору и подготовке предварительных входных данных, которые необходимы для расчета значений выбранных метрик надежности и соответствующих им математических моделей.

Когда значения всех выбранных метрик будут вычислены, на их основе можно выполнить комплексную оценку надежности программного обеспечения. Жизненный цикл программного обеспечения с учетом процесса оценки надежности представлен последовательно параллельной системой и представлен на рисунке 2.

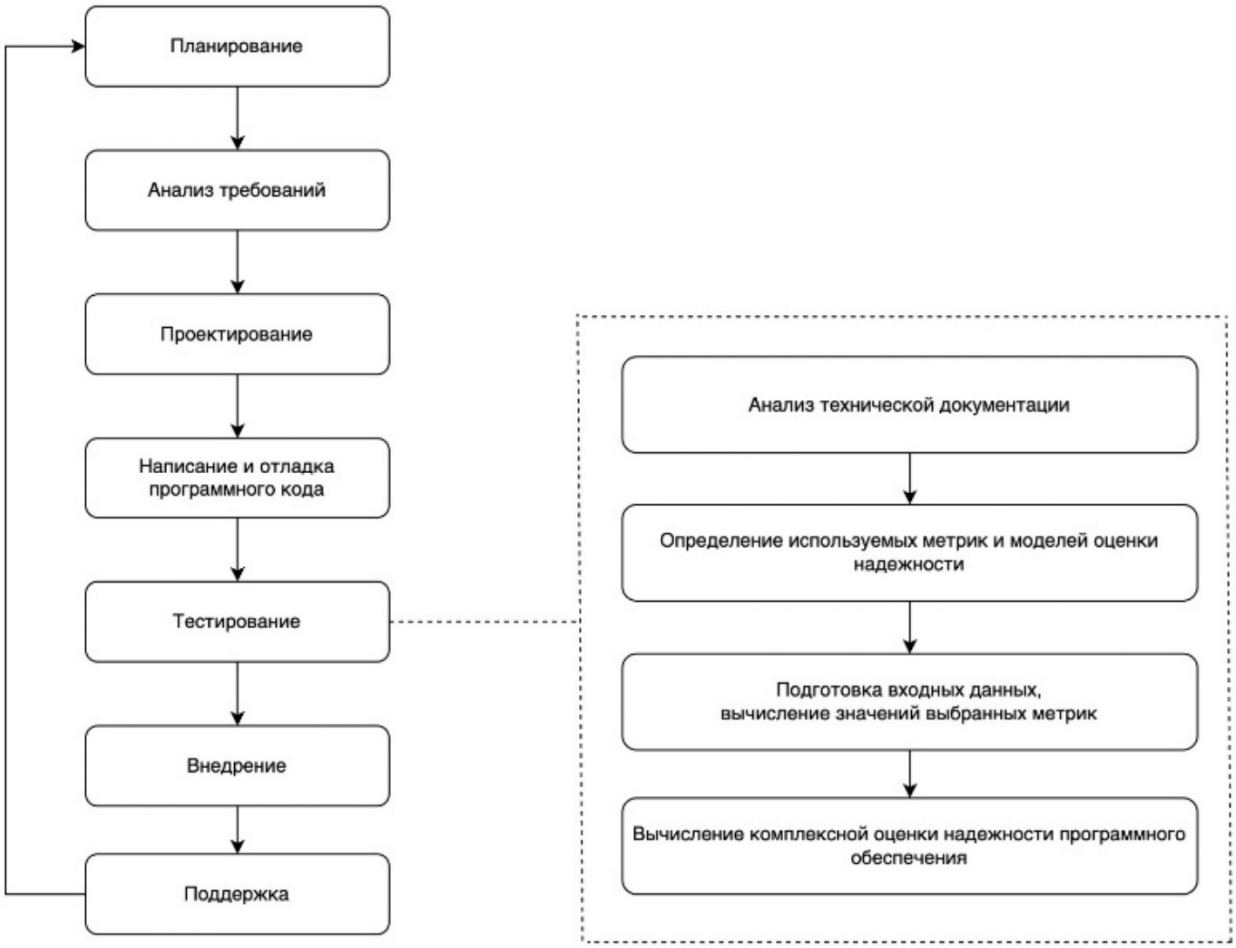


Рис. 2. Жизненный цикл программного обеспечения с учетом процесса оценки надежности

Самые критичные и трудозатратные на исправление ошибки в информационных системах допускаются на первых стадиях жизненного цикла ПО. При неверном планировании времени запрашиваемый функционал может быть не протестирован в должной мере к установленному сроку. Исправление ошибки, возникшей на этапе анализа требований или проектирования, может вызвать необходимость в реорганизации архитектуры программного обеспечения, что спровоцирует изменения во всей информационной системе. Для избежания этих проблем тестирование должно выполняться на каждом этапе жизненного цикла ПО. В связи с данным утверждением необходимо адаптировать процесс оценки надежности всей системы. Жизненный цикл программного обеспечения с учетом процесса оценки надежности на каждом этапе жизненного цикла ПО представлен на рисунке 3.

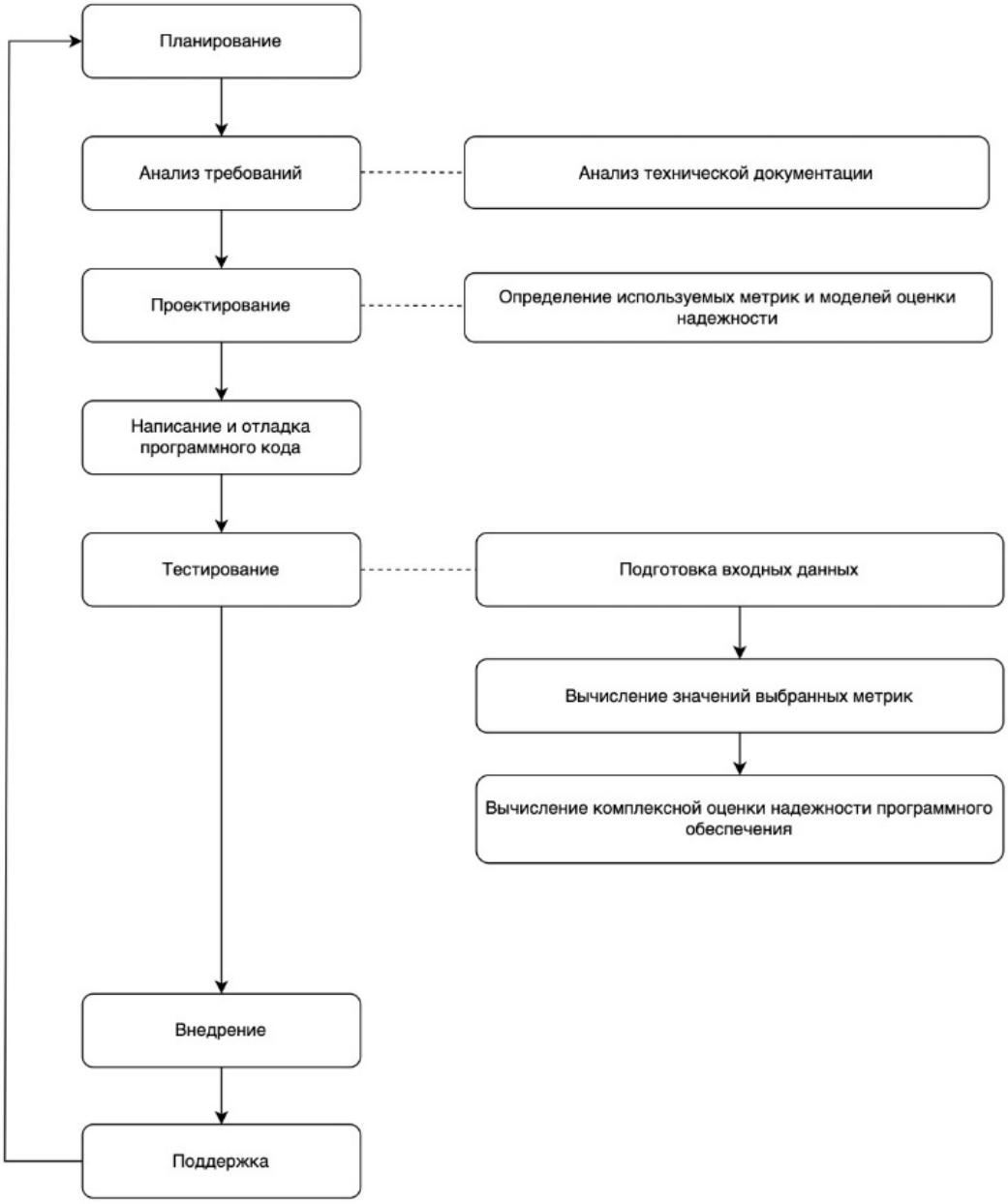


Рис. 3. Жизненный цикл программного обеспечения с учетом процесса оценки надежности на каждом этапе жизненного цикла ПО

В усовершенствованном жизненном цикле ПО некоторым из этапов сопоставлен этап процесса оценки надежности информационной системы.

На этапе анализа требований необходимо проанализировать техническую документацию не только с точки зрения разработки, но и с точки зрения выявления наиболее критичного и важного функционала. Это нужно для определения набора подходящих под конкретную информационную систему метрик на этапе проектирования программного обеспечения.

Для многих математических моделей необходимы заранее известные входные данные [7]. Сбор этой информации и последующее вычисление значений метрик выполняется на стадии тестирования программного обеспечения. На этом же этапе необходимо выполнить комплексную оценку надежности всей информационной системы.

На основе полученной оценке принимается решение о дальнейших действиях – если степень надежности информационной системы достаточно высока, можно переходить к этапу внедрения. Если же уровень надежности программного обеспечения недостаточен – принять меры по ее повышению.

Корреляция результатов значений метрик в оценке надежности программного обеспечения

На этапе тестирования функционала информационной системы выполняется подготовка входных данных и непосредственное вычисление численных значений выбранных метрик надежности информационной системы [8].

Как уже упоминалось ранее, в работах [3,4,5] были выделены следующие базовые метрики, которые могут быть использованы как унифицированный набор метрик для любого программного обеспечения:

- Средняя наработка на отказ;
- Среднее время восстановления;
- Вероятность нахождения в работоспособном состоянии;
- Интенсивность отказов.

Каждой метрике была сопоставлена математическая модель, с помощью которой можно определить ее численное значение.

Знание численных оценок характеристик надежности, как отдельных параметров, не может дать комплексное понимание, в какой мере надежна информационная система. Программа может быть устойчива к отказам и значение параметра нахождения в работоспособном состоянии будет высоким. При этом значение времени восстановления системы может быть настолько велико, что не позволит эффективно использовать данное программное обеспечение. Следовательно, систему нельзя будет назвать надежной и качественной. Поэтому для формирования комплексной оценки надежности информационной системы необходимо учитывать корреляцию всех ее параметров. Взаимосвязь метрик представлена на рисунке 4.

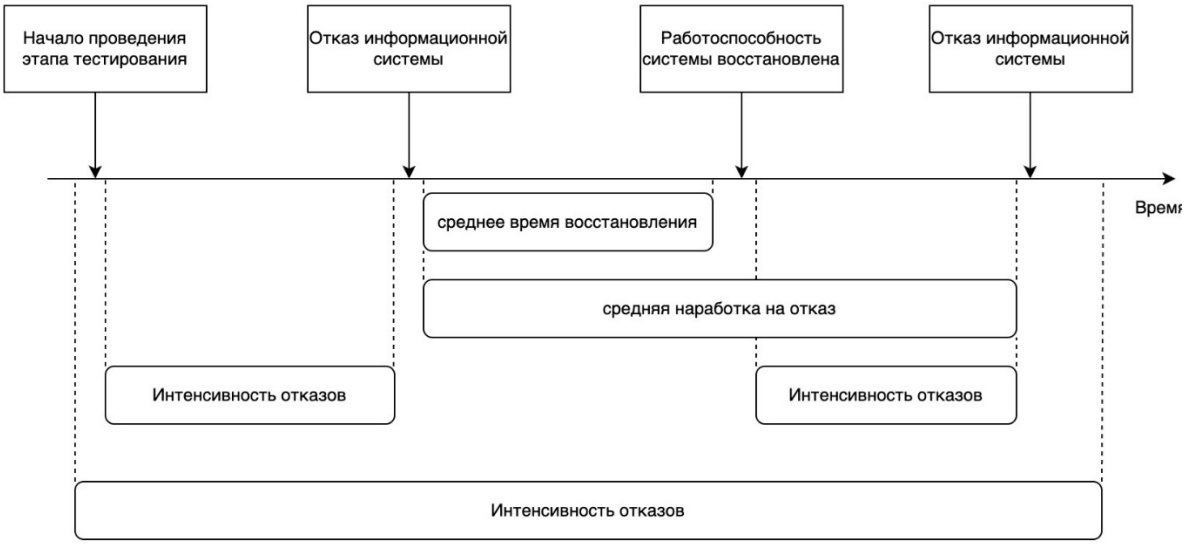


Рис. 4. Взаимосвязь метрик надежности программного обеспечения

Корреляции можно достичь несколькими способами: за счет объединения математических моделей, описывающих параметры надежности, выявления общих величин моделей, а также составления соотношения между значениями отдельных показателей надежности [9]. Одним из способов объединения метрик является составление соотношения между математическими моделями, которые им соответствуют.

Для того, чтобы создать корреляцию между какими-либо параметрами, между ними необходимо выявить что-то общее, через что в дальнейшем будет выстраиваться связь. У всех метрик есть один общий параметр – установленное время тестирования и сбора данных. Следовательно, время может выступать в роли “интерфейса” между метриками надежности.

Если рассматривать надежность информационной системы через призму времени, в этом случае все метрики надежности необходимо выразить через время.

Средняя наработка на отказ и среднее время восстановления уже выражены через время и представляют собой время работы программного обеспечения между двумя последовательными отказами и временем возвращения ПО к полному работоспособному состоянию соответственно. Вероятность нахождения в работоспособном состоянии является противоположностью для метрики средняя наработка на отказ, поэтому может быть выражена так же через время, как обратное от нее значение. Интенсивность отказов характеризуется частотой отказов системы за определенный промежуток времени.

Таким образом, корреляция будет достигаться в том случае, если для расчета каждой из метрик брать один и тот же промежуток времени.

Надежность программного обеспечения, согласно модели Муса, можно выразить по формуле:

$$R = \exp(-\frac{t}{T}), \tag{2}$$

где t – суммарное время функционирования; T – средняя наработка на отказ.

При условии, что промежуток времени равен для всех рассматриваемых моделей выражение для расчета надежности по модели Муса можно преобразовать по формуле:

$$P = \exp(-\frac{t * \lambda}{(F - T) * R}), \tag{3}$$

где t – общее время функционирования программного обеспечения; F – средняя наработка на отказ; T – среднее время восстановления; R – вероятность нахождения в работоспособном состоянии; λ – интенсивность отказов.

Для того, чтобы понять, соответствует ли полученная оценка ожидаемому уровню надежности, необходимо установить граничные значения на основе значений каждого из параметров. Граничные значения для метрик «средняя наработка на отказ», «среднее время восстановления» и «интенсивность отказов» необходимо устанавливать для каждой информационной системы отдельно, основываясь на назначении, степени сложности, области применения и других бизнес-параметров программного обеспечения.

Вероятность нахождения в работоспособном состоянии должна быть максимально приближена к единице [10]. Следовательно, для каждой из информационной системы значение надежности должно устанавливаться свое, учитывая все особенности и специфики данного продукта.

Итеративность оценки надежности программного обеспечения

На первой итерации тестирования должный уровень качества никогда не достигается [11]. На данном этапе всегда обнаруживаются ошибки разной степени важности и критичности. После того, как ошибки зафиксированы, специалисты по тестированию отправляют отчет о них в отдел разработки, где инженеры разработчики исправляют найденные ошибки. После внесения правок в программный код и повторной его отладки снова необходимо провести итерацию тестирования, причем не только того функционала, которого затронули изменения, но и всей системы в целом. Это необходимо в связи с тем, что изменения в системе могли затронуть другие ее модули, что создает возможность для возникновения новых ошибок.

Количество таких итераций зависит от количества найденных ошибках и степени их влияния на качество всей информационной системы. Так как надежность зависит и от количества ошибок в программном коде, на первой итерации тестирования ожидаемая степень надежности так же не будет достигнута. Аналогично найденным ошибкам, результаты оценки надежности должны быть проанализированы и улучшены на последующем этапе. График достижения должного уровня надежности информационной системы отражен на рисунке 5.

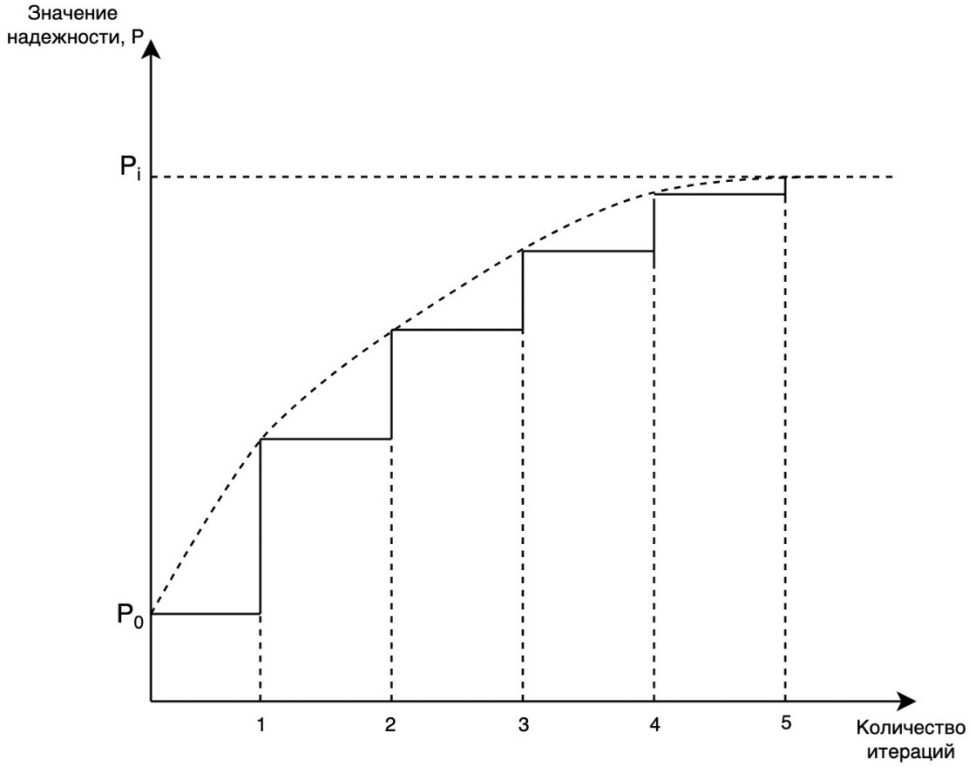


Рис. 5. График достижения ожидаемого уровня надежности

Влияние надежности на качество программного обеспечения

Для того, чтобы понять, какого рода ошибки могут повлиять уровень надежности программного обеспечения, в понятии надежности можно выявить следующие составляющие:

- Корректность функционирования;
- Недоступность данных;
- Устойчивость работы.

Корректность функционирования – свойство программы удовлетворять ее функциональному предназначению [13]. Корректность подразумевает полное соответствие функциональным требованиям к продукту, соответствие ожидаемого поведения программы фактическому. Функциональные требования обеспечивают четкое описание того, как система должна реагировать на конкретную команду, функции и ожидания пользователей. Функция представляет собой действие пользователя: ввод данных, нажатие на кнопку, переход по страницам и прочее, и реакция системы на данное действие. Корректность функционирования обеспечивается на итерациях тестирования программного обеспечения.

Недоступность данных является мерой безопасности как для системы, так и для компании, в которой будет эксплуатироваться данное программное обеспечение. Недоступность достигается за счет сокрытия части функционала для определенных пользователей, а также сокрытие данных, которые принадлежат другим пользователям. Обычно недоступность реализуется посредством разделения ролей и учетных записей пользователей в системе и необходима в целях обеспечения корпоративной безопасности.

Также стоит обеспечить недоступность данных на уровне системы в рамках самой системы. К примеру, если у пользователя каким-либо образом получится изменить данные в тот момент, когда их будет менять система, это может привести к некорректным выходным данным или даже отказу программного обеспечения. Разделение ролей и сокрытие данных на уровне системы также можно проверить на этапе тестирования заранее подготовленными тестовыми сценариями.

Устойчивость работы программного обеспечения – свойство программы во время эксплуатации не быть чувствительной к ошибкам, которые не были найдены во время итераций тестирования, отказам и внешним факторам [12]. Это свойство аналогично свойству отказоустойчивости аппаратуры. К факторам, которые могут повлиять на устойчивость работы, можно отнести внешние факторы при эксплуатации системы. К примеру, конфликт с другими программными обеспечениями на компьютере пользователя – данные ПО могут обращаться к одному участку памяти, или использовать большое количество ресурсов операционной системы, что не даст второму программному обеспечению безотказно функционировать, при этом ошибок такое ПО может и не содержать. Также на устойчивость могут повлиять некорректные действия пользователя или искажение входной информации. Для того, чтобы избежать ошибок, информационная система должна содержать в программном коде методы обработки исключений и, по возможности, наличие подсказок на интерфейсе, чтобы скорректировать действия пользователя и предотвратить отказ всей системы или искажения других данных.

Главным отличием устойчивости от других составляющих надежности является невозможность тестирования до передачи системы в эксплуатацию. Это связано с тем, что сценарии отсутствия устойчивости можно воспроизвести либо на протяжении длительного времени тестирования, на что нет времени и ресурсов во время разработки ПО, либо условия возникновения подобных сценариев настолько специфичны, поэтому их всех предусмотреть нельзя.

Устойчивость программного обеспечения характеризуется количеством отказов за определенный промежуток времени. Чем меньшее количество раз система отказала, тем она устойчивее к вышеперечисленным факторам. Следовательно, разработку программного обеспечения необходимо выстроить таким образом, чтобы максимально были учтены всевозможные внешние факторы.

Можно выделить несколько принципов разработки программного обеспечения, для повышения не только устойчивости всей системы, но и как следствие, повышения уровня надежности:

– Рациональный выбор алгоритмического языка, позволяющего сократить число операций и команд при выполнении поставленной задачи;

– Введение в разрабатываемые программы тестовых задач, позволяющих обнаружить сбои и отказы в работе программ и даже произвести автоматическое переключение на резерв или на повторение операций;

– Определение места возникновения неисправности (диагностические тесты) с целью ускорения работ по устранению неисправностей;

– Обеспечение постоянной готовности резервных устройств к использованию их в качестве основных;

– Обеспечение возможности перестроения структуры комплекса на новый режим работы в случае отказа некоторой его части (выполнение всего объема задач комплекса одной вычислительной машиной, хотя и с меньшим быстродействием, изменение приоритетности обработки данных), переход на другие средства отображения информации и т.д.

Однако, во время процесса написания и отладки кода необходимо учитывать время и способы восстановления системы после отказа. К примеру, в следствии возникновения специфичной и трудновоспроизводимой ошибки в системе возникает отказ, после чего система перезагружается без потери данных в течении нескольких секунд. При этом на исправления данной ошибки необходимо потратить много ресурсов, а ошибка может больше и не появиться. Тогда на надежности и на качестве системы в целом такая ситуация не сказывается, что отражает формула:

$$K >> T - \frac{T_\delta}{Q}, \tag{4}$$

где K – время, которое необходимо потратить на исправление ошибки, вызывающей отказ; T – время безотказной работы программного обеспечения; T_δ – время восстановления программы; Q – средняя наработка на отказ.

Следовательно, если время, которое необходимо потратить на исправление ошибки, вызывающей отказ, сильно больше, чем разница между временем корректного функционирования и временем на восстановление работы, править такую ошибку не целесообразно.

Заключение

При оценке надежности конкретной информационной системы прежде всего важно подобрать наиболее подходящие метрики. Для того, чтобы это сделать необходимо модифицировать процесс разработки программного обеспечения с учетом этапов определения набора подходящих метрик надежности, подготовки входных данных для их расчета, а также этапа расчета корреляционной оценки.

Этап расчета корреляционной оценки является одним из наиболее важных этапов, так как позволяет подвести итог и сказать, в какой мере надежна информационная система и можно ли передавать ее в эксплуатацию пользователю. При этом по отдельным значениям метрик корректно оценить надежность системы невозможно, так как все метрики взаимосвязаны между собой. Поэтому в работе было предложено решение объединить модели, которые описывают метрики, через соотношение. Объединяющим параметром для каждой из модели было выбрано время.

Так как надежность зависит и от количества ошибок в программном коде, на первой итерации тестирования ожидаемая степень надежности так же не будет достигнута. Этот факт говорит о том, что достижение ожидаемого показателя надежности информационной системы является итеративным процессом, и никогда не достигается на первом этапе.

Одним из способов повышения надежности информационной системы является сокращение количества ошибок в программном коде. При этом важно помнить, что, если время, которое необходимо потратить на исправление ошибки, вызывающей отказ, сильно больше, чем разница между временем корректного функционирования и временем на восстановление работы, править такую ошибку не целесообразно.

Литература

1. *Гадасин Д.В., Шведов А.В., Ермалович А.В*. Концепция "туманные вычисления" – эволюционный этап развития инфокоммуникационных технологий // Технологии информационного общества: Сборник трудов XII Международной отраслевой научно-технической конференции, Москва, 14-15 марта 2018 года. Том 2. М.: Издательский дом Медиа Паблишер, 2018. С. 96-99. EDN XUPRRB.

2. *Гадасин Д.В., Шведов А.В., Ермалович А.В*. Концепция "Интернет вещей" как вектор развития информационно-коммуникационных технологий на пути к "Индустрии 4.0" // Технологии информационного общества: XI Международная отраслевая научно-техническая конференция: сборник трудов, Москва, 15-16 марта 2017 года. М.: Издательский дом Медиа паблишер, 2017. С. 352-353. EDN RPDYWX.

3. *Яковенко Н.В., Коровушкина В.М*. Классификация моделей оценки надежности программного обеспечения // Актуальные проблемы и перспективы развития экономики : Труды XXI Международной научно-практической конференции, Симферополь-Гурзуф, 20-22 октября 2022 года. Симферополь: Крымский федеральный университет им. В.И. Вернадского, 2022. С. 292-294. EDN RTXKWR.

4. *Докучаев В.А., Шведов А.В.* Классификация показателей надежности корпоративных цифровых платформ // Актуальные проблемы и перспективы развития экономики : труды XIX Всероссийской с международным участием научнопрактической конференции, Симферополь-Гурзуф, 15-17 октября 2020 года. Симферополь: ИП Зуева Т. В., 2020. С. 28-29. EDN NFEHDJ.

5. *Липаев В.В*. Надежность программных средств. М.: СИН-ТЕГ, 1998. 232 с.

6. *Maklachkova V.V., Shvedov A.V., Alyev S.* Analysis of Resilience Indicators in Corporate Networks and Possible Ways to Improve It // 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2022 – Conference Proceedings, Moscow, 15-17 марта 2022 года. Moscow, 2022. DOI 10.1109/IEEECONF53456.2022.9744353. EDN ZRMVJS.

7. *Гадасин Д.В., Гадасин В.А.* Система комплексной оценки информационных технологий // Технологии информационного общества : X Международная отраслевая научно-техническая конференция: сборник трудов, Москва, 16-17 марта 2016 года. М.: Издательский дом Медиа паблишер, 2016. С. 18-21. EDN VPDXOX.

8. *Шведов А.В.* Повышение эффективности функционирования корпоративных информационно-коммуникационных сетей с учетом теории ограничения систем // III Научный форум телекоммуникации: теория и технологии ТТТ-2019 : Материалы XXI Международной научно-технической конференции, Казань, 18-22 ноября 2019 года. Том 1. Казань: Казанский государственный технический университет им. А.Н. Туполева, 2019. С. 290-291. EDN NJLIPU.

9. *Гадасин В.А., Гадасин Д.В.* Стратегическая надежность информационных 3D-мегасистем // Телекоммуникационные и вычислительные системы : Труды конференции, Москва, 24 ноября 2015 года. М.: ООО "Брис-М", 2015. С. 38-39. EDN VRLZAZ.

10. *Shvedov A.V., Nazarov M.J.* Methods for improving the efficiency of information and communication networks // 2020 International Conference on Engineering Management of Communication and Technology, EMCTECH 2020 - Proceedings, Vienna, 20-22 октября 2020 года. Vienna, 2020. P. 9261563. DOI 10.1109/EMCTECH49634.2020.9261563. – EDN ZVEXSW.

11. *Суслин А.А.* Экспериментальное исследование взаимосвязи значений метрик и показателей надежности программного обеспечения // Молодой ученый. 2010. № 6. С. 67-71. EDN MUASVX.

12. *Zolotukhin P.A., Melkova E.K., Gadasin D.V., Korovushkina V.M.* Using Intelligent Testing as a Tool to Improve the Quality of Transport Information Systems // 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2022 - Conference Proceedings, Moscow, 15-17 марта 2022 года. Moscow, 2022. DOI 10.1109/IEEECONF53456.2022.9744348. EDN NOMJLX.

13. *Шведов А.В., Гадасин Д.В., Коровушкина В.М., Мелькова Е.К.* Интеллектуальное тестирование как способ повышения качества информационной системы // REDS: Телекоммуникационные устройства и системы. 2022. Т. 12. № 2. С. 43-52. EDN GOLZGE.

ОБЗОР ТЕХНОЛОГИИ DOCKER

Едлин Виталий Аркадьевич,
Московский технический университет связи и информатики, Москва, Россия,
edlin84@mail.ru

Аннотация
В статье рассматриваются вопросы виртуализации в целом и контейнеризации в частности. Ключевым объектом рассмотрения является архитектура Docker. В рамках работы освещаются технологии, на основе которых построена платформа Docker, ее особенности, отличие от классической контейнеризации, а также плюсы и минусы использования контейнеризации на уровне операционной системы и на уровне приложений.

Ключевые слова: *контейнеризация, виртуализация, Docker, контейнер Линукс, разработка*

Введение

Прежде чем затронуть основную тему работы, необходимо отметить, что Docker является крайне популярной программой / платформой, активно применяющейся в практической разработке и размещении приложений. Применительно к тому, как работает Docker, сложно найти тему, которая не была бы так или иначе освещена в документации [1], научных работах или публицистических материалах [1-14].

Не ставя перед собой задачи выявить что-либо новое в творении разума и рук человеческих на нынешнем этапе собственного познания Docker'а, мной предлагается лишь краткий обзор технологии.

Итак, поскольку начинать принято с начала, мы начнем с описания.

Docker можно определить как программное обеспечение, предоставляющее удобное средство для создания и запуска контейнеров, а также управления ими. Сами разработчики, определяя Docker, смещают акцент с технологии в сторону продаваемого результата и комплексной инфраструктуры, поддерживаемой как крупными компаниями, так и сообществом: это открытая платформа для разработки, доставки и запуска приложений [1]. Действительно, в настоящее время Docker представляет собой не только механизм запуска и управления контейнерами, но и огромную библиотеку образов, а также множество дополнительных возможностей, облегчающих разработку и внедрение (доставку) программ. По своей сути контейнеры не являются чем-то новым, а сам Docker построен на известной технологии разделения пространства имен и cgroups [1], которая активно применяется в операционных системах семейства Linux.

Как известно, технология может быть сколь угодно замечательной, но ее успех зависит от того, насколько она проста и понятна рядовому потребителю услуги. Docker обеспечил такие удобства, предоставив разработчикам несколько более адаптированный механизм технологии контейнеризации, чем были предложены ранее, и стал таким образом одним из элементов стандартного набора технологий, задействованных в разработке программного обеспечения.

Абстракции и виртуализация

Вопрос виртуализации тесно связан с абстрагированием: над оборудованием; над оборудованием и управляющей системой; над оборудованием, гипервизором, операционной системой, отдельными ресурсами операционной системы и проч. Слоев (уровней) этой абстракции может быть очень много.

Исторически потребность в виртуализации возникла в связи с необходимостью обеспечить соответствие между неравномерным развитием аппаратного и программного обеспечения. Очевидно, что каждый раз переписывать программное обеспечение под изменившееся оборудование было крайне трудозатратным.

Можно сказать, что вся работа с компьютерными системами строится на композиции и формировании на их основе абстракций. Композиции позволяют нам создавать из множества атомарных (в рамках соответствующего уровня) элементов новые целостные элементы, которые будут атомарными на более

высоких уровнях создания композиций и абстрагирования. Абстракция как механизм, освобождающий нас от необходимости анализа не актуальных в данном случае свойств объекта, позволяет использовать результат композиции как атомарный элемент, не учитывая сложные процессы, происходящие в каждом элементе, из которых он состоит. Нам необходимо лишь чтобы на каждом уровне абстракции нам был предоставлен интерфейс для получения необходимых услуг (сервиса). При этом то, что происходит «под капотом» уже не должно занимать наше внимание.

Например, рассматривая распределенные системы, исследователи [2 с. 133] отмечали, что «каждая (распределенная) компьютерная система предлагает программный интерфейс для программного обеспечения более высокого уровня». По своей сути виртуализация позволяла заменить один интерфейс на другой, понятный на более высоком уровне, имитируя поведение системы на более низком уровне.

Исследователи [2 с. 135] выделяют следующие уровни интерфейсов:

1) интерфейс между аппаратным и программным обеспечением (архитектура набора инструкций), включающий в себя привилегированные и общие инструкции;

2) интерфейс системных вызовов операционной системы;

3) интерфейс библиотечных вызовов (API).

Виртуализация предполагает имитацию поведения этих интерфейсов.

Архитектура компьютера как «железа» содержит несколько уровней абстракции, упрощая понимание происходящих в нем процессов и управление ими. Операционная система также является уровнем абстракции, а ее возможности позволяют нам формировать все новые и новые уровни. По справедливому замечанию Э. Таненбаума, «абстракция является ключом к управлению сложностью. Хорошая абстракция превращает практически неподъемную задачу в две, решить которую вполне по силам. Первая из этих задач состоит в определении и реализации абстракций, а вторая – в использовании этих абстракций для решения текущей проблемы» [3 с. 25].

В некоторых случаях нам необходимо работать с полноценной операционной системой, для чего мы создаем виртуальную машину с полноценной гостевой системой. В зависимости от реализации такая виртуализация может поддерживаться аппаратно или программно. В других случаях нам необходимо несколько изолированных полноценных операционных систем. Для определенных целей нам может быть не потребуются все возможности операционной системы, но будет нужно лишь легковесное изолированное окружение для запуска и работы пользовательской программы. В иной ситуации нам необходимо представить управляющей программе фасад якобы целостного аппаратного обеспечения, в то время как ресурсы могут быть конструктивно отнесены к отдельным устройствам.

В каждом из описанных случаев в принципе можно говорить о виртуализации, поскольку в любом случае от «железа» нас отделяет несколько уровней абстракции, представляющих нам удобные интерфейсы для работы.

Так или иначе, виртуализация позволяет нам создавать на базе массива физических ресурсов множество изолированных пространств с различными характеристиками. Как правило, эти характеристики сводятся к объему функциональных возможностей – от полновесной операционной системы, «верящей», что она одна у этого «железа», до отдельного гостевого изолированного процесса, обращающегося к ядру хост-системы и «умирающему» после завершения запущенной в контейнере программы.

Если посмотреть на виртуализацию под иным углом, то можно заметить, что ключевой задачей является максимально эффективное использование аппаратного обеспечения. А как этого добиться? Э.Таненбаум
[3 с. 531] выделил три требования, которым должны соответствовать гипервизоры:

1) безопасность;

2) эквивалентность;

3) эффективность.

Несмотря на то, что эти принципы касались гипервизоров, на взгляд автора они касаются виртуализации в целом. Так, нам необходимы системы, которые обеспечивали бы:

1) эффективность, которая выражалась бы в использовании ресурсов «железа» по максимуму. Это может выражаться с одной стороны в возможности создания минимально необходимых для решения задач систем, позволяющих получать доступ к ресурсам компьютера, а с другой стороны в увеличении их количества на единицу аппаратного обеспечения.

2) безопасность – на взгляд автора, здесь речь идет, прежде всего, об изоляции как на аппаратном, так и на программном уровне в зависимости от типа виртуализации и используемой технологии. Мы

можем изолировать гостевые операционные системы, взаимодействующие с гипервизором; можем изолировать среды исполнения, помещая их в контейнеры, представляющие собой смонтированные файловые системы с собственным пространством имен, а также использовать иные технологии.

3) эквивалентность – все то, что было обозначено в п.п. 1 и 2, не должно снижать требуемый уровень качества работы, в том числе по сравнению с системой управления ресурсами компьютера, установленной на отдельном оборудовании и располагающей доступом к ресурсам оборудования без не естественных для этой системы «прослоек».

От поставленных целей и характера решаемых задач зависит баланс указанных характеристик и, как следствие, избранная технология виртуализации. Выделяют [4] следующие типы виртуализации:

1) аппаратная виртуализация: эмуляция;

2) программная виртуализация: динамическая трансляция и паравиртуализация;

3) виртуализация на уровне операционной системы: контейнеры.

Каждый из этих типов по своей сути смещает описанный выше баланс в ту или иную сторону. Разумеется, можно выделить и другие типы виртуализации, в том числе в зависимости от того, к чему они применяются. В данном случае мы применяем ранее описанную абстракцию.

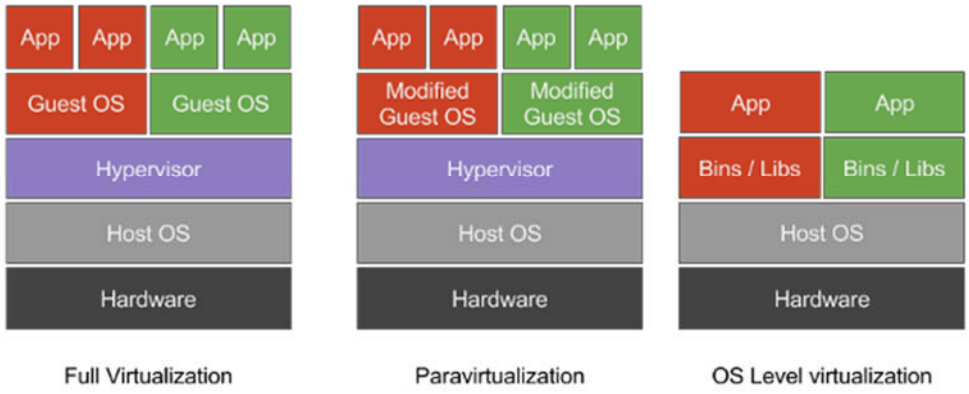


Рис. 1. Разница между полной виртуализацией, паравиртуализацией и контейнеризацией [5]

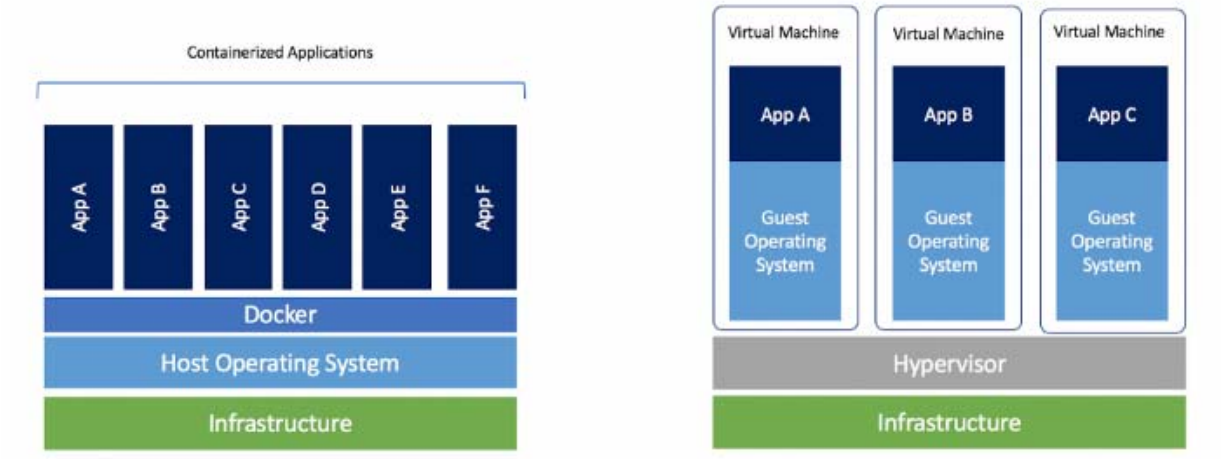


Рис. 2. Разница между виртуализацией и контейнеризацией в контексте Docker [5]

Контейнеризация

Контейнеризация не является новой технологией, но представляет собой один из типов виртуализации. В ряде работ можно встретить противопоставление виртуализации и контейнеризации [6, 7, 8] на концептуальном уровне. Однако фактически в таком противопоставлении виртуализация связывается, прежде всего, с виртуальными машинами, которые представляют собой программно или программно-аппаратно созданное виртуальное железо «над» которым работает гостевая операционная система.

Однако виртуализация – это не только виртуальные машины, в связи с чем формальное обозначение противопоставления представляется не корректным. В то же время противопоставление виртуальных машин и контейнеров видится вполне оправданным и обоснованным.

Отличительной чертой контейнеризации, которая как раз позволяет противопоставить ее виртуальным машинам, является то, что контейнеры опираются на одно общее ядро операционной системы [2 с. 169]. При таких обстоятельствах управление осуществляется операционной системой хоста.

Можно сказать, что контейнеризация как технология является естественной для архитектуры UNIX-систем, а возможные нововведения, обеспечивающие наилучшую безопасность и изолированность, оперативно внедрялись и включались в ядро Linux.

В качестве основы контейнеров чаще всего рассматривают операцию chroot, характерную для UNIX-систем. Данная операция позволяет изменить корневой каталог /. По сути, такая операция позволяет создать некую «песочницу» для работы в новом пространстве, к которому должны быть примонтированы необходимые каталоги и устройства. В то же время chroot обладает рядом недостатков, связанных с безопасностью, в том числе с возможностью выйти из «песочницы» или запускать из нее процессы во внешней части системы.

Таким образом, по своей природе контейнер – это изолированный процесс (группа процессов), запущенные на одной хостовой операционной системе. В связи с этим некоторые исследователи выделяют следующие определяющие особенности контейнеров [9]:

- выполнение на одном хосте;
- группа процессов, исходящих из корневого процесса, изолированного в контейнере;
- изолированность контейнеров;
- выполнение этой группой процессов общих функций.

Собственно, дальнейшее развитие контейнеризации было построено на обеспечении наиболее безопасного и эффективного использования данной технологии. Так, вместо chroot используется pivot_root. Для полноценного функционирования контейнера необходима соответствующая файловая система (rootfs), содержащая необходимые библиотеки.

В целях обеспечения изоляции была внедрена технология пространства имен (namespaces). Здесь мы тоже говорим об абстракции, которая создает иллюзию того, что все происходит не в контейнере, а в действительном корне системы. В частности, создается пространство имен для файловой системы, ID процессов, сети, пользовательских ID и проч.

Так, например, пространство имен ID процессов допускает ситуацию, когда процессы в контейнере и на хосте имеют один и тот же ID, что не нарушает порядок работы. Первый процесс, создаваемый в контейнере, имеет номер 1 и обрабатывается так же, как и обычный процесс инициализации. При этом сам этот контейнерный 1 процесс также может создавать свои процессы по тому же принципу.

Аналогичная ситуация и с user ID: ID пользователя в контейнере может совпадать с ID пользователя на хосте, однако это будут два разных пользователя.

Следующей особенностью является контрольная группа (cgroups), разработанная в недрах Google и включенная в ядро Linux в 2008 г. Контрольная группа представляет собой механизм изоляции отдельных процессов (процессорные, сетевые, память, ввод-вывод) и управление ими в соответствии с иерархией групп [10]. Данный механизм предусматривает [11] ядро cgroups core и ряд подсистем, которые и определяют различные ограничения. Например, devices определяет порядок доступа к устройствам, cpu обеспечивает доступ процессов к процессору и проч.

Таким образом, связка namespaces & cgroups позволила, с одной стороны, выделить пространство имен для поддержания иллюзии единоначалия процессов в контейнере, а с другой стороны определила ограничения для использования процессами в группах ресурсов компьютера.

Указанные технологии поступательно включались в ядро Linux, формируя обеспечение использования контейнеров Linux (LXC). Контейнеры LXC позволяют смонтировать несколько экземпляров операционной системы на базе хостовой операционной системы. При этом каждый такой экземпляр будет изолирован, иметь свое пространство имен и ограничения по действиям, но при этом использовать единое ядро базовой операционной системы, в чем заключается ключевое отличие от виртуальных машин. Иными словами, LXC не формирует виртуальную машину, но лишь создает необходимое для выполнения задач окружение. То есть смонтированный как контейнер экземпляр операционной системы содержит лишь необходимые для выполнения этих задач инструменты (библиотеки, сетевое

окружение и проч.), что обеспечивает легковесность таких контейнеров с одной стороны и относительно ограниченный функционал с другой.

Основы технологии Docker

Итак, что же предоставляет нам Docker? Все вышесказанное по поводу контейнеров имеет непосредственное отношение к архитектуре Docker. Дело в том, что Docker до 2014 г. использовал в качестве своей основы контейнеры LXC. С 2014 г. в версии 0.9 Docker начал использовать собственную разработку – libcontainer [12]. Почему так произошло и зачем было нужно?

Libcontainer позволяет получить некоторую независимость от LXC и обеспечивает совместимость с другими инструментами изоляции, в том числе libvert и system-nspawn [13]. При этом libcontainer в своей работе использует технологии namespaces и cgroups, а также apparmor, chroot и проч.

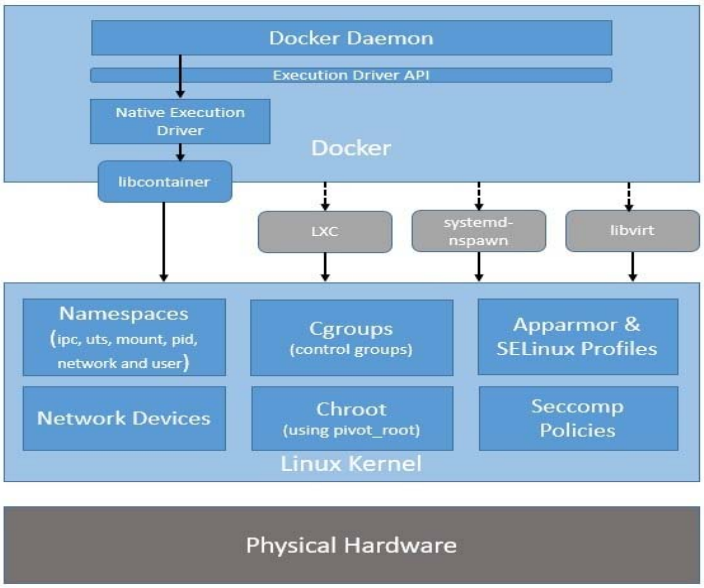


Рис. 3. Использование Docker с libcontainer [13]

Такой подход был направлен на исключение зависимости от внешних источников вроде LXC при сохранении технологий виртуализации и изоляции контейнеров. Следует отметить, что Docker активно участвует в стандартизации контейнеризации, так что отказ от LXC представляется вполне логичным.

Отмечается, что архитектура Docker неоднократно менялась, но на настоящий момент [13] выглядит следующим образом (рис. 4).

Docker работает по принципу взаимодействия клиента и сервера. Клиент предоставляет интерфейс командной строки (CLI) или REST API, позволяющий взаимодействовать с сервером – Docker Daemon. Клиент-серверная архитектура позволяет разнести управляющую и управляемую структуру в пространстве.

Следующим аспектом архитектуры Docker является использование образов (images). Именно образы являются ключевым фактором, обеспечивающим переносимость контейнеров и удобство при сборке приложений.

Контейнеры Docker запускают процессы, определенные образами. Образы состоят из нескольких слоев и некоторых метаданных Docker, обеспечивающих работы на платформе. Сами образы рассматривают как слои файловой системы, доступные только для чтения и включающие в себя различные библиотеки и другие зависимости, необходимые для нормальной работы приложения. В качестве абстракции для понимания соотношения образов и контейнером предлагается [14 с. 27] рассматривать первые как программы, а вторые как процессы либо как классы и объекты соответственно (если применять аналогию с ООП).

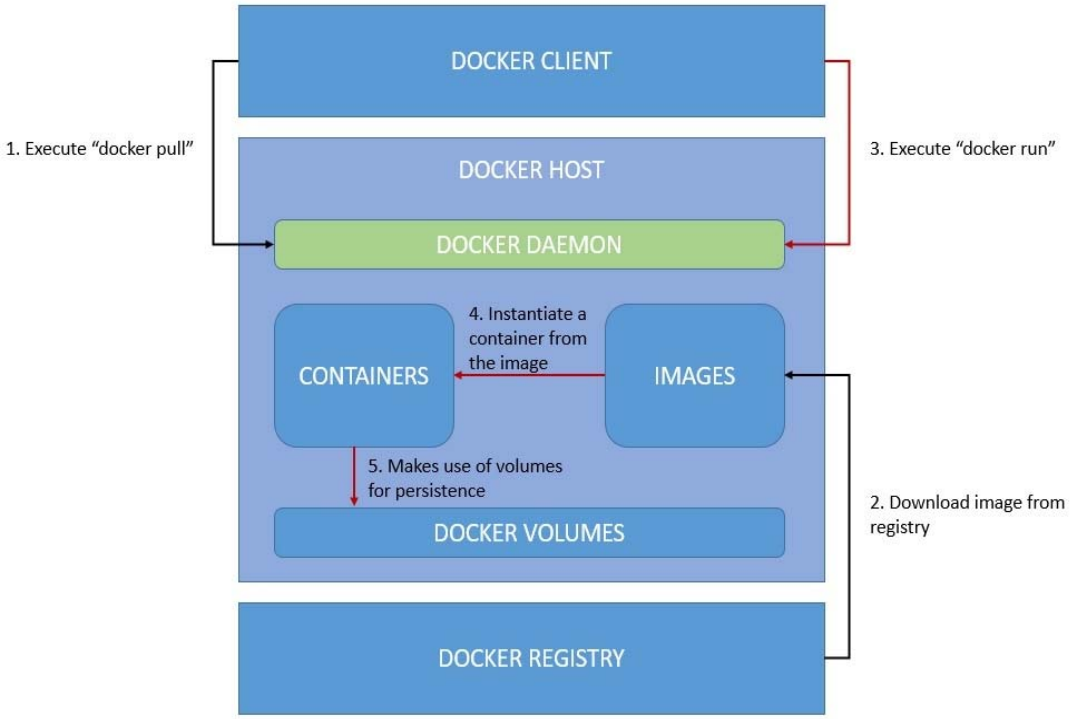


Рис. 4. Архитектура Docker с libcontainer. Не включает в себя создание образа и публикацию в реестре [13]

В свою очередь слои – это наборы изменений в файлах. Контейнер Docker – это запущенный экземпляр образа. Аналогия с классом и объектом выражается в том, что может быть запущено несколько контейнеров с одним образом. Такие контейнеры будут изолированы друг от друга, а изменения в контейнере не повлияют на сам образ, что не исключает создание нового образа из измененных слоев.

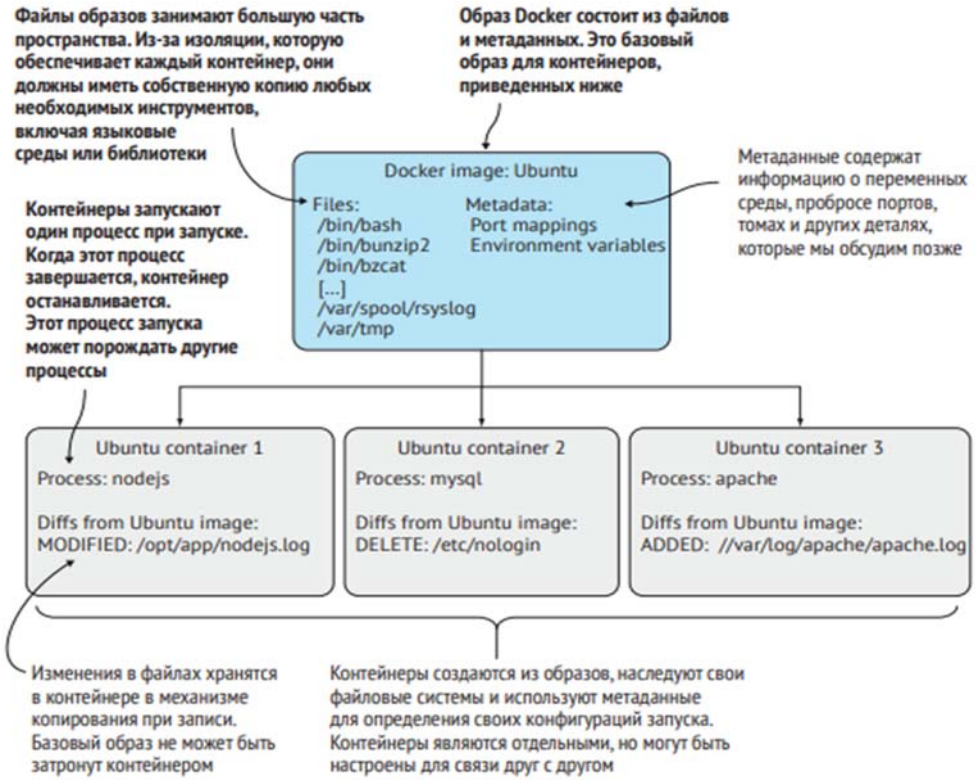


Рис. 5. Образы и контейнеры Docker [14 с. 28]

Особенностью слоев является то, что все изменения по сравнению с первоначальным образом сохраняются не в виде нового набора файлов, а с использованием технологии копирования при записи [14 с. 37]: при создании нового объекта из образа не требуется копировать весь набор данных, так как достаточно скопировать только сами изменения.

Дело в том, что слои сформированного образа доступны только для чтения. При формировании контейнера из сформированного образа также создается слой для чтения и записи, который позволяет учитывать изменения. При запуске контейнера создается пространство процессов вокруг данного контейнера.

Собственно, при использовании команды docker create <> из образа, где файловая система предполагает только чтение, создается контейнер с областью для чтения и записи. При использовании команды docker start <> происходит запуск контейнера – в нем появляется пространство процессов.

Когда мы используем команду docker ps, мы получаем на вывод перечень существующих процессов, то есть запущенных контейнеров. При использовании команды docker ps –a на вывод подается перечень всех контейнеров, в том числе не запущенных.

При использовании команды docker stop <> процесс контейнера завершается, у контейнера больше нет пространства процессов. При удалении контейнера (docker rm <>) удаляется слой для чтения и записи.

Ниже приведен шаблонный пример создания, запуска и удаления контейнеров, а также образов. Стоит иметь в виду. Что команда docker run <> сначала ищет образ в локальном репозитории, а при отсутствии такового осуществляет его загрузку, создание и запуск контейнера. Таким образом, команда run скрывает «под капотом» несколько операций, рассмотренных нами ранее.

```
vitaly@vitaly-VirtualBox:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:c77be1d3a47d0caf71a82dd893ee61ce01f32fc758031a6ec4cf1389248bb833
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Рис. 6. Запускаем контейнер hello-world. Как видно, локально образ не был найден, в связи с чем был произведен поиск в библиотеке образов

```
vitaly@vitaly-VirtualBox:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
vitaly@vitaly-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                      PORTS     NAMES
b46d4dcb30f4   hello-world   "/hello"   21 seconds ago   Exited (0) 18 seconds ago             unruffled_turing
```

Рис. 7. Просматриваем активные процессы (docker ps) и все процессы (docker ps –a). Как видим, контейнер hello-world закончил свою работу и процесс, составляющий данный контейнер, был прекращен.

```
vitaly@vitaly-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED              STATUS                          PORTS     NAMES
f69d87fb3d0a   hello-world   "/hello"   14 seconds ago       Exited (0) 13 seconds ago                 dazzling_
stonebraker
b46d4dcb30f4   hello-world   "/hello"   About a minute ago   Exited (0) About a minute ago             unruffled
_turing
```

Рис. 8. Как видим, при повторном запуске контейнера hello-world, был создан новый контейнер, который, отработав, закончил процесс

```
vitaly@vitaly-VirtualBox:~$ sudo docker rm dazzling_stonebraker
dazzling_stonebraker
vitaly@vitaly-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED         STATUS                     PORTS     NAMES
b46d4dcb30f4   hello-world   "/hello"   2 minutes ago   Exited (0) 2 minutes ago             unruffled_turing
vitaly@vitaly-VirtualBox:~$ sudo docker rm unruffled_turing
unruffled_turing
vitaly@vitaly-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Рис. 9. Удаляем контейнеры

```
vitaly@vitaly-VirtualBox:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    feb5d9fea6a5   15 months ago   13.3kB
vitaly@vitaly-VirtualBox:~$ sudo docker images -a
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    feb5d9fea6a5   15 months ago   13.3kB
vitaly@vitaly-VirtualBox:~$ sudo docker rmi feb
Untagged: hello-world:latest
Untagged: hello-world@sha256:c77be1d3a47d0caf71a82dd893ee61ce01f32fc758031a6ec4cf1389248bb833
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
Deleted: sha256:e07ee1baac5fae6a26f30cabfe54a36d3402f96afda318fe0a96cec4ca393359
vitaly@vitaly-VirtualBox:~$ sudo docker images -a
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
vitaly@vitaly-VirtualBox:~$
```

Рис. 10. Как видим, при использовании команды docker run hello-world из репозитория был локально загружен образ hello-world. С помощью команды docker rmi hello-world образ был удален

Дополнительно стоит отметить, что, действительно, для работы Docker использует root права, что может негативно сказаться на безопасности.

Одной из отличительных черт Docker является то, что LXC, как было указано выше, предполагает упаковку в контейнер части операционной системы и необходимые приложения. В свою очередь контейнеризация Docker предлагает возможность создать контейнер приложений, под который будет создаваться самостоятельный процесс [5].

Контейнер с ОС: - запускается как ОС; - запускает несколько сервисов в одном контейнере; - использует встроенные механизмы изоляции ресурсов (Linux) **Примеры:** - LXC - OpenVZ - Linux-VServer - Solaris Containers	**Контейнер на уровне приложения:** - запускается как изолированное приложение; - запускает каждый процесс в собственном контейнере; - является надстройкой над контейнерами на уровне ОС **Примеры:** - Docker - RKT

Таким образом, вместо одного относительно «тяжелого» контейнера с операционной системой и стеком приложений для работы мы получаем несколько контейнеров, которые взаимодействуют между собой. Такой подход представляется вполне актуальным для конструирования микросервисов, поскольку позволяет выстраивать архитектуру, состоящую из относительно «легких» составляющих, обеспечивая масштабируемость проектов. В свою очередь, сама платформа Docker позволяет управлять жизненным циклом контейнеров. При этом следует учитывать, что поскольку каждый запущенный контейнер представляет собой процесс, по завершению этого процесса контейнер прекращает работу.

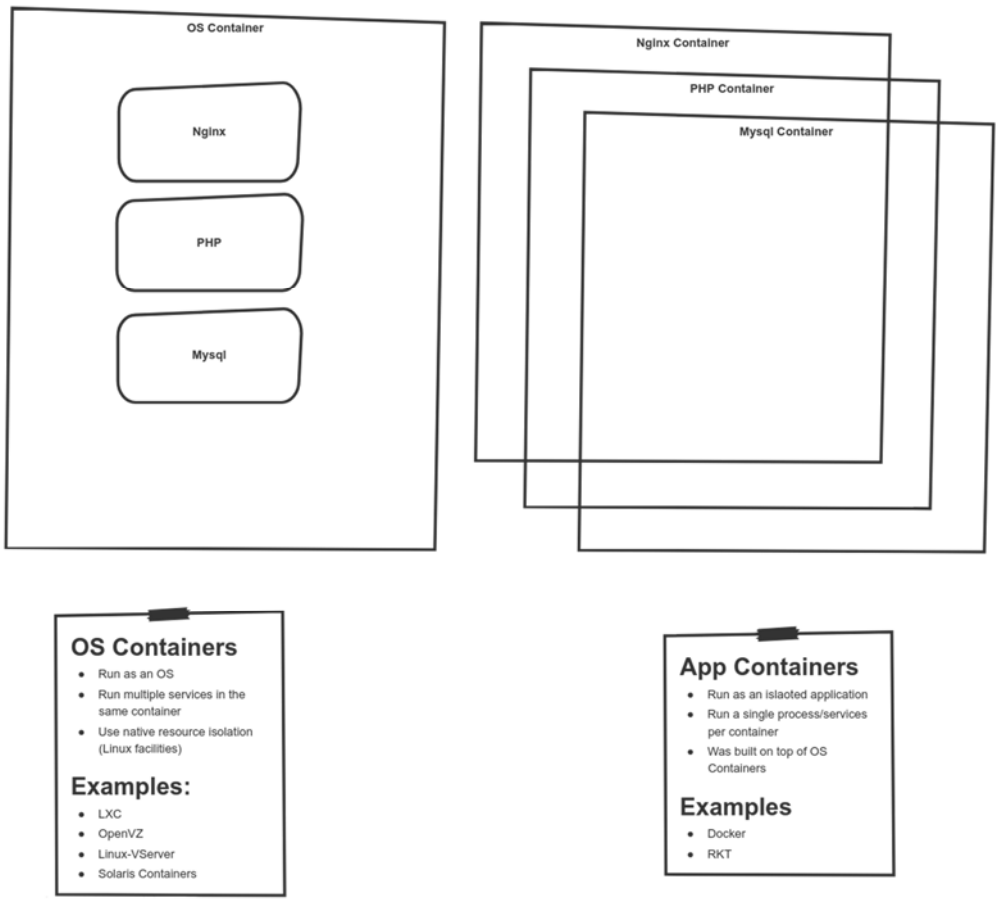


Рис. 9. Разница между контейнерами на уровне операционной системы (LXC, OpenVZ и проч.) и контейнерами на уровне приложения (Docker). Также здесь показана «слоеная» структура [5]

Контейнер LXC предоставляет возможность изоляции процессов, однако не гарантирует переносимость контейнеров [13]. В свою очередь Docker предоставляет потребителю легкую переносимость контейнеров. В частности, такая переносимость обеспечивается рассмотренной выше технологией формирования слоев и извлечения образов.

Эти плюсы в некоторых случаях могут выступать и как минусы платформы. Так, отмечают следующие недостатки контейнеров Docker [6]:

- контейнеры рассчитаны, прежде всего, на размещение в них относительно небольших программ или отдельных частей больших программ, которые будут взаимодействовать между собой. Например, отдельные контейнеры для БД Postgres, для серверов Nginx и Apache, для самого приложения с необходимыми для его запуска и работы окружения и зависимостями;

- совместимость. Отмечается, что контейнеры не всегда могут взаимодействовать между собой. Учитывая то, что именно эта задача часто выступает приоритетной для контейнеров, недостаток может оказаться существенным;

- ограничение жизненного цикла контейнера. Контейнер запускается как процесс, действующий внутри контейнера. Как только процесс внутри контейнера завершается, контейнер «умирает». С одной стороны, для решения такого вопроса существуют различные методы, но с другой стороны представляется, что это вполне оправданная цена для такой технологии, а само прекращение функционирования контейнеров с завершением процесса является вполне адекватным концептуальным решением;

- есть мнение, что безопасность Docker обеспечивается в меньшей степени, чем в LXC [15], поскольку демон Docker управляет объектами (контейнеры, сети, образы и проч.), а также выполняет запросы API Docker как root на хосте.

Выводы

Фактически Docker стал стандартом в контейнеризации. Для тех целей, когда не требуется использование всех возможностей виртуальных машин, Docker предложил легковесные изолированные контейнеры, содержащие все, что необходимо для запуска приложений. Собственно, это касается контейнеризации в целом: это и возможность удобного прототипирования программного обеспечения, создание «песочницы» и проч. Наряду с этим Docker предоставил широкие возможности масштабирования программных продуктов, а легкие контейнеры, которых можно создать в большом количестве и организовать взаимодействие между ними, позволяет реализовать архитектуру микросервисов.

Создание необходимых для запуска образов, включающих все необходимые зависимости, стало как никогда простым и понятным: пошаговая инструкция в Dockerfile позволила воспроизводить программное окружение из готовых образов в короткие сроки.

Несмотря на то, что базово Docker был направлен на работу исключительно с ядром Linux, в настоящее время он активно применяется на Windows. Для этого используется инструмент WSL (Windows Subsistem for Linux). Также Docker представил удобное управление с помощью Desktop версии как для Linux, так и для Windows / macOS.

Экосистема Docker представляет потребителям готовые образы (images) приложений (Docker Hub), которые можно использовать в качестве шаблонов для формирования контейнеров. Само создание образов активно поддерживается сообществом, в связи с чем можно найти необходимый образ для решения огромного количества задач. Также все больше инструментов поддерживает интеграцию с Docker, в т.ч. Jenkins, Travis и иные инструменты CI/CD.

Возможность конструирования контейнеров на основе Dockerfile позволяет наладить процессы непрерывной интеграции и непрерывного развертывания (CI/CD) без использования сторонних инструментов.

В целом, Docker как раз и ориентирован на сферу DevOps, предоставляя готовые решения, необходимые в этой части процесса разработки. Docker сформировал готовый продукт, предоставляющий не просто технологию, но целый конвейер для решения рутинных задач. Такой продукт оказался актуальным для многих разработчиков, чем снискал популярность и стал фактически стандартом для контейнеризации на уровне приложений.

По мнению специалистов [10] Docker хорош в большей степени как контейнер для упаковки одного процесса или службы, в то время как LXC можно представить как «монолит», включающий в одну среду все зависимости для работы приложения. Docker хорошо справляется с собиранием окружения из готовых образов и формированием слоев, в то время, как LXC хорош, когда нет желания создавать большое количество слоев и контейнеров, а концепция приложения строится на размещении всех сервисов в одном контейнере.

Таким образом, вряд ли можно сказать, что контейнеры на уровне приложений являются эволюцией по сравнению с контейнерами LXC. Представляется, что в данном случае можно говорить о конкретизации задач и поиске для них самого оптимального решения сообразно рассмотренным ранее принципам безопасности, эквивалентности и эффективности.

Заключение

Происходящее в информационных технологиях подчиняется принципу, сформулированному Г. Гегелем, согласно которому достижение определенного объема количественных изменений приводит к качественным изменениям: происходит выход за пределы меры, то есть предела количественных изменений в рамках данного качества.

Идеи виртуализации не новы, а количественные изменения в технологиях рано или поздно приводят к революциям. Такие революции, конечно же добавляя что-то новое (новую технологию или новое применение старой технологии), используют как плацдарм то качество, где происходили количественные изменения. Идеи абстракции и композиции в совокупности с требованиями потребителей породили виртуализацию.

Различные типы виртуализации формировались под влиянием изменяющихся потребностей и соответствующего им баланса между безопасностью, эффективностью и эквивалентностью.

Очевидно, что для одних задач оптимальным будет использование виртуальных машин, для других – «классических» контейнеров на уровне операционной системы, а для третьих – «слоеных» контейнеров на уровне приложений. Добавляя к этому экосистемы разнообразных проектов и инструментов, адаптированных для решения конкретных задач, мы увидим большое количество изменений. Нет сомнений в том, что при преодолении такими количественными изменениями критического объема, произойдут качественные изменения.

Литература

1. Docker Documentation [Электронный ресурс]. URL: https://docs.docker.com/ (дата обращения: 22.01.2023).
2. *Стинван М., Таненбаум Э.С*. Распределенные системы. М: ДМК Пресс, 2021. 584 с.
3. *Таненбаум Э., Бос Х.* Современные операционные системы. 4-е изд. СПб.: Питер, 2020. 1120 с.
4. Виртуализация [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Виртуализация (дата обращения: 22.01.2023).
5. Недостающее введение в контейнеризацию / il_da_r [Электронный ресурс]. URL: https://habr.com/ru/post/541288/ (дата обращения: 22.01.2023).
6. Виртуализация и контейнеризация: обзор технологий и в чем разница [Электронный ресурс]. URL: https://dzen.ru/a/Y5CtLYpgfj2WJXQX (дата обращения: 22.01.2023).
7. Visualization vs Containerization [Электронный ресурс]. URL: https://www.baeldung.com/cs/virtualization-vs-containerization (дата обращения: 22.01.2023).
8. Контейнеризация понятным языком: от самых азов до тонкостей работы с Kubernetes / freetonik [Электронный ресурс]. URL: https://habr.com/ru/company/southbridge/blog/530226/ (дата обращения: 22.01.2023).
9. *Sascha Grunert.* Demystifying Containers – Part I: Kernel Space [Электронный ресурс]. URL: https://medium.com/@saschagrunert/demystifying-containers-part-i-kernel-space-2c53d6979504 (дата обращения: 22.01.2023).
10. Контейнер LXC для веб-разработки как альтернатива Docker / agorlov [Электронный ресурс]. URL: https://habr.com/ru/post/563040/ (дата обращения: 22.01.2023).
11. *Andrei Yemelianov*. Механизмы контейнеризации: cgroups [Электронный ресурс]. URL: https://habr.com/ru/company/selectel/blog/303190/ (дата обращения: 22.01.2023).
12. *Solomon Hykes*. Docker 0.9: introducing execution drivers and libcontainer [Электронный ресурс]. URL: https://www.docker.com/blog/docker-0-9-introducing-execution-drivers-and-libcontainer/ (дата обращения: 22.01.2023).
13. *Kumar Chandrakant*. Evolution of Docker from Linux Containers [Электронный ресурс]. URL: https://www.baeldung.com/linux/docker-containers-evolution (дата обращения: 22.01.2023).
14. *Иан Милл, Эйдан Хобсон Сейерс*. Docker на практике. М.: ДМК Пресс, 2020. 516 с.
15. *Eric Kahuha*. LXC vs Docker: Which Container Platform Is Right for you [Электронный ресурс]. URL: https://earthly.dev/blog/lxc-vs-docker/ (дата обращения: 22.01.2023).

ИССЛЕДОВАНИЕ И РАЗРАБОТКА КОМПЛЕКСНОЙ СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ НА ПРЕДПРИЯТИИ СФЕРЫ ОПТОВОЙ ТОРГОВЛИ

Кузиков Сергей Никодимович,
МТУСИ, Москва, Россия,
tank32@mail.ru

Воронова Лилия Ивановна,
МТУСИ, зав.кафедрой ИСУиА, д.ф-м.н., Москва, Россия
voronova.lilia@ya.ru

Аннотация
В статье исследуется степень обеспечения информационной безопасности предприятия оптовой торговли, применение комплексного подхода к обеспечению информационной безопасности на предприятии. Рассматриваются средства и методы обеспечения информационной безопасности на предприятии. Описывается результаты применения разработанного комплекса защиты информационной безопасности на предприятии оптовой торговли ООО «БИТ».

Ключевые слова: *информационная безопасность, комплексная система защиты информации, несанкционированный доступ, программно-аппаратные меры защиты информации, информационные системы*

Введение

С бурным развитием программных и технических средств, появлением персональных компьютеров, развитием глобальных сетей, появлением сети интернет, умных устройств, технических средств разведки конфиденциальной информации проблема защиты информации обострилась.

Информационные угрозы в современном обществе включают в себя огромное разнообразие всех категорий нарушений конфиденциальности, целостности и доступности информации. За последние годы киберпреступники постоянно меняют способы, средства и методы для получения нужных данных.

Каждодневная работа предприятия связанна большим объемом обрабатываемой информации: работа с электронными документами, обработка конфиденциальных данных. Это включает в себя большое количество рисков и угроз для целостности обрабатываемой информации. С каждым годом объемы обрабатываемых данных увеличиваются, как следствие, растёт и количество вероятных угроз.

Для предотвращения всего комплекса угроз, как внешних, так и внутренних или большей его части, необходимо рассматривать проблему защиты информации на предприятии, используя комплексный подход с использованием современных технических и программных средств.

Результаты исследования

КСЗИ – комплексная система защиты информации предприятия – сочетание набора средств, методов и мер (организационных, технических, правовых) используемых для недопущения несанкционированного доступа к информации, обеспечивающих эффективную защиту информации на предприятии [1]. Схема реализации КСЗИ на предприятии представлен на рисунке 1.

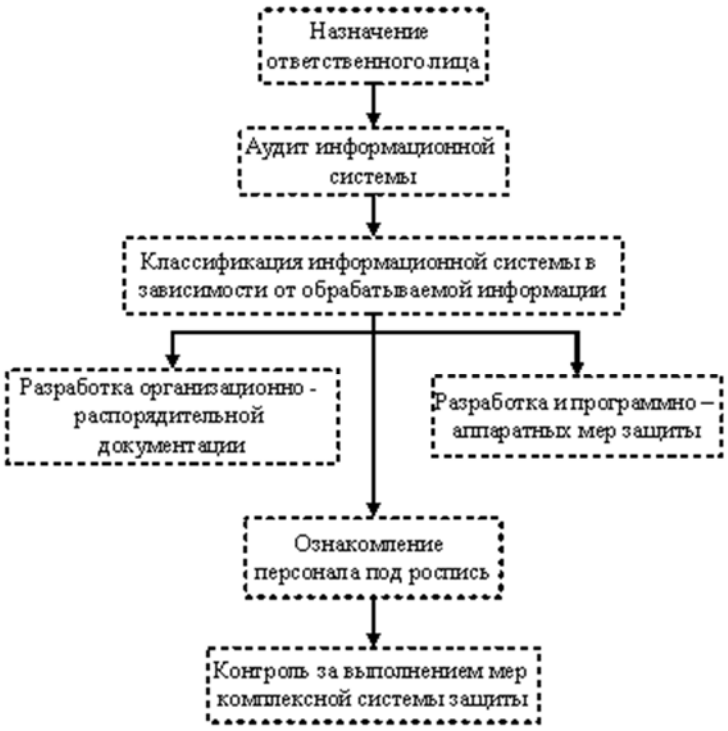


Рис. 1. Схема реализации КСЗИ на предприятии

Для реализации организационных мер по обеспечению ИБ на предприятиях оптовой торговли должен быть выполнен ряд мероприятий по: организации контроля учета рабочего времени; системы контроля и управления доступом (СКУД); подбору персонала; организации видеонаблюдения; регламентированию парольной защиты, ИБ, доступа в технические помещения; организации физической охраны; назначение ответственного за защиту информации, разработке организационно-распорядительной документации[2].

При разработке организационно-распорядительной документации, следует учитывать особенности основной деятельности предприятия и регламентирующей:

- Цели и задачи обеспечения информационной безопасности на предприятии, описание основных угроз безопасности, проводимые организационные и технические мероприятия по обеспечению информационной безопасности, состав и структуру автоматизированной системы управления, а также состав средств защиты;
- Правила безопасной работы и повышения осведомленности работников, включая действия при возникновении нештатных ситуаций;
- Планы мероприятий по обеспечению информационной безопасности (для каждой АСУ предприятия может быть разработан свой план, учитывающий специфику процессов), порядок реализации отдельных мер обеспечения информационной безопасности и реагирования на нештатные ситуации, компьютерные инциденты, порядок информирования и обучения работников предприятия.

Подробный состав и формы организационно-распорядительной документации каждое предприятие определяет с учетом специфики своей деятельности [3].

Вся организационно-распорядительная документация должна быть доведена до сведения подразделений и лиц, обеспечивающих ИБ на предприятии, в части, их касающейся, а также до руководства данного предприятия.

Проанализировав текущее требование законодательства, возможно отметить следующие акты и нормативно-правовую документацию необходимых при разработке организационно-распорядительной документации, регламентирующей политику информационной безопасности организации:

- Федеральный закон от 26 июля 2017 г. № 187-ФЗ;
- Приказ ФСТЭК России от 06 декабря 2017 № 227 [4];
- Постановление Правительства РФ от 8 февраля 2018 г. № 127;

- Приказ ФСТЭК России от 11 декабря 2017 № 229 [5];
- Приказ ФСТЭК России от 21 декабря 2017 № 235 [6, 7];
- Приказ ФСТЭК России от 22 декабря 2017 № 236 [8, 9];
- Приказ ФСТЭК России от 25 декабря 2017 г. № 239 [10];
- Приказ ФСТЭК России от 9 августа 2018 № 138[11];
- Методический документ «Методика оценки угроз безопасности информации» (утв. ФСТЭК России 5 февраля 2021 г.)» [12];
- Банк данных угроз безопасности информации ФСТЭК [13];
- Проект указа Президента РФ «О мерах экономического характера по обеспечению технологической независимости и безопасности объектов критической информационной инфраструктуры» [14].

В ходе анализа вышеуказанных законодательных актов и нормативно-методических документов, выделены основные:

- Федеральный закон от 26 июля 2017 г. N 187-ФЗ "О безопасности критической информационной инфраструктуры Российской Федерации"

Под действие Федерального закона N 187-ФЗ как значимые объекты КИИ подпадают многие предприятия промышленности РФ [15, 16]. Основными сферами деятельности закона являются: наука, финансы, связь, промышленность, транспорт, здравоохранение, оборонно-промышленный комплекс.

- Федеральный закон от 26 июля 2017 г. N 187-ФЗ кроме дисциплинарной, гражданско-правовой и административной ответственностью предусматривает и уголовную ответственность за нарушение требований безопасности КИИ [17,18].
- Постановление Правительства РФ от 8 февраля 2018 г. N 127 «Об утверждении Правил категорирования объектов критической информационной инфраструктуры Российской Федерации, а также перечня показателей критериев значимости объектов критической информационной инфраструктуры Российской Федерации и их значений»
- Постановление Правительства РФ N 127 [19, 20] утверждает правила присвоения категорий, путем детального описания процесса категорирования, с указанием критериев и методики расчета данных критериев. Существует также отдельная отраслевая версия данного документа для топливно-энергетического комплекса РФ [21].
- Приказ ФСТЭК России от 21 декабря 2017 N 235 «Об утверждении Требований к созданию систем безопасности значимых объектов КИИ РФ и обеспечению их функционирования» содержит требования по обеспечению защиты объектов КИИ на этапе создания их систем безопасности [6, 7].
- Приказ ФСТЭК России от 22 декабря 2017 N 236 «Об утверждении формы направления сведений о результатах присвоения объекту КИИ одной из категорий значимости либо об отсутствии необходимости присвоения ему одной из таких категорий» устанавливает форму предоставления данных касательно присвоении или об отсутствии необходимости присвоения объекту КИИ категории значимости. При формировании данного документа необходимо указать все без исключения виды компьютерных инцидентов, которые могут произойти в следствии осуществления угроз ИБ, а также целенаправленных атак или предоставить сведения о невозможности осуществления таких угроз ИБ [8, 9]. К целенаправленным компьютерным атакам относятся: отказ в обслуживании; несанкционированный доступ; утечка данных (нарушение конфиденциальности); модификация (подмена) данных; нарушение функционирования технических средств; несанкционированное использование вычислительных ресурсов объекта.
- Приказ ФСТЭК России от 25 декабря 2017 г. N 239 «Об утверждении Требований по обеспечению безопасности значимых объектов КИИ РФ» – обеспечение безопасности в ходе создания, эксплуатации и вывода из эксплуатации значимых объектов КИИ. [10, 22, 23].
- «Методика оценки угроз безопасности информации» (утв. ФСТЭК России 5 февраля 2021 г.) содержит очередность и содержание работ по выявлению угроз безопасности информации (УБИ). С целью оценки угроз ИБ в системах и сетях применяется положение данной методики и распространяется на все созданные, модернизированные системы и сети решение, о которых принято после даты ее утверждения, а также для всех существующих [12].

- Банк данных УБИ [13]. Содержащиеся в Банке данных, не являются полной и может быть дополнены по результатам анализа угроз безопасности информации и уязвимостей в конкретной ИС (АСУ) с учетом ее особенностей и эксплуатации.

Организационные и нормативно-правовые методы так же регламентируют порядок внедрения технических средств защиты информации. Комплексный подход к выбору и сочетанию различных технических и программных средств обеспечит необходимый уровень защиты информации.

Программные и технические средства, обеспечивающие защиту информации на предприятии: firewall (МСЭ), антивирусное программное обеспечение, программные средства диагностики и мониторинга сетевых компьютеров, криптографическое ПО.

Аудит ИТ-инфраструктуры предприятия оптовой торговли ООО «БИТ» показал полное отсутствие регламентирующей документации, политики ИБ, программных средств антивирусной защиты, МСЭ, СКУД, видеонаблюдения, физической охраны, ответственных лиц.

Критически важные ИС, маршрутизаторы, коммутаторы, сервера баз данных, сервер AD - находятся в одной подсети, отсутствует разграничение управляющих портов.

Удаленное подключение филиалов осуществляется по протоколу PPTP - один из старейших VPN протоколов, используемых до сих пор.

Обнаружено использование нелицензионного ПО: 98% программного обеспечения Microsoft не лицензировано. На АРМ установлены не лицензионные программные продукты компаний Adobe, Abbyy, Autodesk, Corel, RarLab.

По результатам работы сетевого сканера выявлены уязвимости в пограничном оборудовании, ip-телефонии, АРМ.

Процедуры бэкап на критически важных узлах информационной инфраструктуры – отсутствуют.

В программном обеспечение выявлены эксплуатируемые уязвимости в том числе 0-day.

В маршрутизаторах выявлены уязвимости в том числе устаревшее ПО.

Анализ трафика показал присутствие ботов, вирусов, спам роботов и пр. на PC и серверах компании. На рисунке 2 представлена выдержка из отчета по сканированию трафика в ЛВС предприятия, в котором отражены наиболее серьезные атаки, зафиксированные в сети.

Protection Name	Severity
Suspicious Executable Mail Attachment	Critical
Adobe Reader PDF CIDFont Dictionary Memory Corruption (APSB11-16)	Critical
SIPVicious Security Scanner	High
Cisco Multiple Products Denial of Service (CVE-2018-15454)	High
Microsoft Windows RDP Brute Force Login Attempt	High
Memcached Web-Servers Network Flood Denial of Service	High
Unicom Suspicious Evasion Technique	High
Shodan Scanner SIP Request	High

Рис. 2. Отчет по сканированию трафика ЛВС ООО «БИТ»

Количество атак с использованием уязвимостей, представленных на рисунке 2, представлен на рисунке 3.

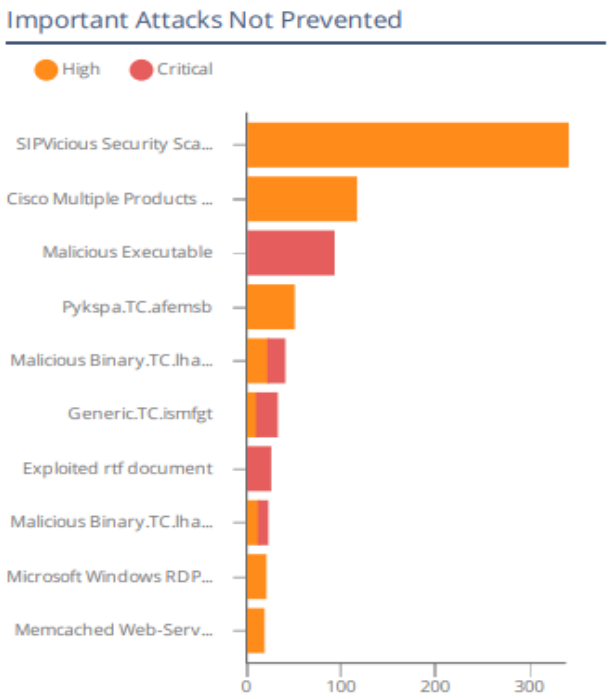


Рис. 3. Отчет по количеству атак в день на предприятии ООО «БИТ»

На рисунке 4 выдержка из отчета по использованию потенциально опасных приложений в сети предприятия.

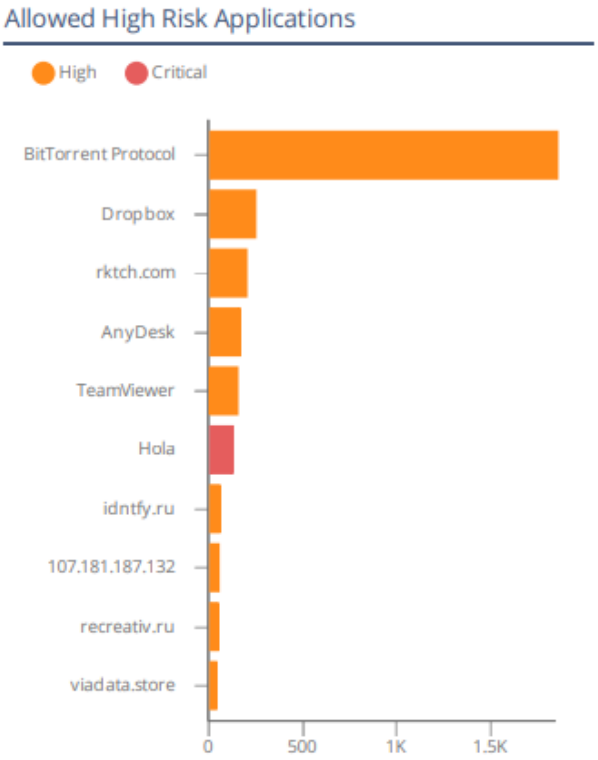


Рис. 4. Потенциально опасные приложения в сети предприятия ООО «БИТ»

Анализ сетевого трафика проводился с помощью специализированного ПО компании CheckPoint Endpoint security [24]. Check Point Endpoint Security – программное обеспечение с централизованным управлением для защиты корпоративной сети и входящих в её состав компьютеров, и серверов, обеспечивающее защиту от вирусов и угроз безопасности.

Полные результаты аудита обобщены и изложены автором в корпоративном документе «Отчет по аудиту ИТ-инфраструктуры предприятия оптовой торговли ООО «БИТ»», являющемся основой для принятия решения изменения ИТ-инфраструктуры.

Разработанный автором комплекс обеспечения ИБ, в виде применения организационных мер (организационно-распорядительная документация), организацией СКУД в комплексе с системой видеонаблюдения с модулем искусственного интеллекта и технических средств на базе Cisco ASA, который был успешно реализован в автоматизированной системе управления, функционирующей на предприятии оптовой торговли.

Организована системы контроля и управления доступом (СКУД) в комплексе с системой видеонаблюдения с модулем искусственного интеллекта (распознавание лиц). На рисунке 5 изображена структурная схема организации СКУД в центральном офисе ООО «БИТ».

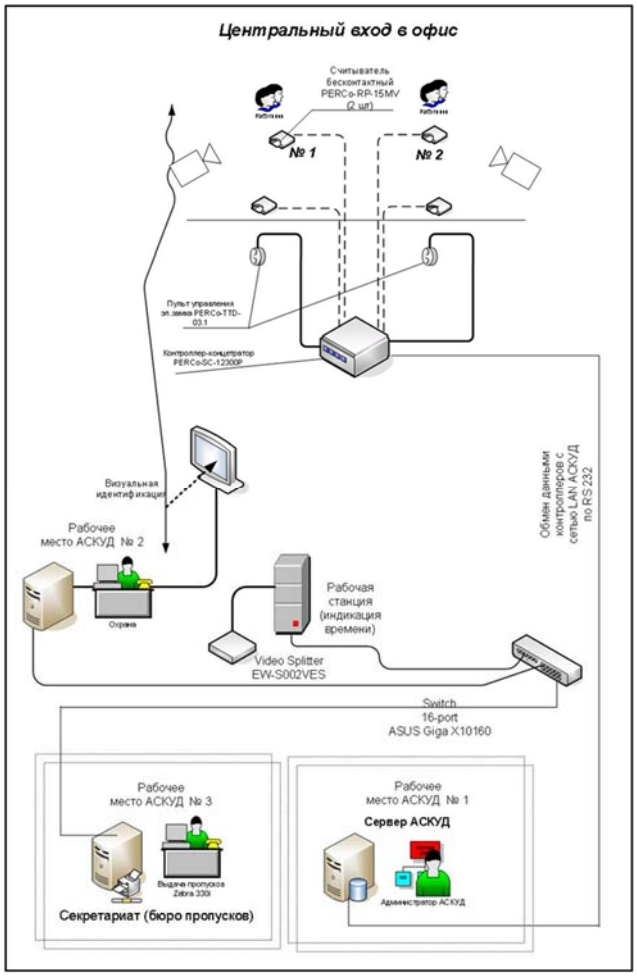


Рис. 5. Структурная схема организации СКУД на предприятии ООО «БИТ»

Система видеонаблюдения установлена в центральном офисе и всех филиалах компании с единым центром управления. Подключение удаленных IP-камер осуществляется по второстепенному каналу связи, организованном в каждом филиале, реализованным по технологии VPN site-to-site IPSec [25], что гарантирует шифрование траффика (передаваемые данные защищены от нарушения целостности и достоверности).

Управление IP-камерами будет осуществляться путем специализированного ПО на АРМ сотрудника путем прохождения процедуры идентификации и аутентификации посредством электронной подписи.

По итогам разработанной КСЗИ, предложенной автором, в течении 8 месяцев на предприятии оптовой торговли ООО «БИТ» проходила поэтапная замена серверов и АРМ задействованных для критически важных процессов компании. В результате проведенной работы в парке предприятия обновлено 98% техники, включая сервера, АРМ, периферийные и сетевые устройства.

Внедрен Dr.Web Enterprise Security Suite – комплекс программных продуктов Dr.Web для защиты корпоративных сетей от всех видов интернет-угроз с Центром управления для удобства администрирования. Инсталлирован антивирусный сервер – компьютер, находящийся в локальной сети предприятия, на котором установлено ПО Dr.Web Enterprise Security Server.

Перечень реализованных в ЛВС центрального офиса средств защиты информации:

cisco SmallBussines SG-200; средство централизованной антивирусной защиты DrWeb, для защиты оконечных устройств применяется DrWeb Security Suit; сетевая система обнаружения атак Cisco FirePower ASA 55XX;

Обслуживаемые системы: межсетевой экран; система обнаружения вторжений.

Средства защиты АРМ: средство антивирусной защиты DrWeb Security Suit.

Перечень реализованных в ЛВС филиалов средств защиты информации: cisco SmallBussines SG-100; средство централизованной антивирусной защиты DrWeb, для защиты оконечных устройств применяется DrWeb Security Suit; сетевая система обнаружения атак Cisco FirePower ASA 551X;

Обслуживаемые системы: межсетевой экран; система обнаружения вторжений.

Средства защиты АРМ: средство антивирусной защиты DrWeb Security Suit.

Описание специализированных систем безопасности, примененных в сети доступа.

Межсетевой экран – Cisco FirePower

Межсетевой экран организован на основе устройства Cisco FirePower ASA 55XX в сети доступа центрального офиса.

Помимо применения аппаратного фаервола в сети организации для обеспечения ИБ реализованы:

Ограничение доступа на уровне портов.

Ограничение доступа осуществляется посредством Port-Security – это функция коммутатора, при помощи которой мы можем указать каким устройствам можно пропускать трафик через определенные порты. Устройство определяется по его MAC-адресу.

Сбор авторизационных и аутентификационных данных осуществляется сервером Cisco.

Фильтрация трафика на сетевом, транспортном уровнях и уровне приложений.

Контроля доступа к внешней сети, приложениям, устройствам и сервисам внутри ЛВС используются списки доступа (Access Control Lists (ACL)). Помимо этого, ЛВС сегментирована, процесс сегментации сети включает в себя разделение физической сети на разные логические подсети, что упрощает управление, повышая уровень контроля сетевого трафика, оптимизируя производительность сети и улучшая систему безопасности.

Организация подключения филиалов Предприятия к ЛВС центрального офиса.

Подключение филиалов ООО «БИТ» осуществляются путем организации VPN site-to-site IPsec, реализованных на оборудовании Cisco asa 551x.

Для реализации организационных мер защиты разработаны следующие организационно-распорядительные документы:

– политика информационной безопасности ООО «БИТ»;

– план мероприятий по обеспечению безопасности ООО «БИТ», включающий правила и процедуры информирования и обучения персонала, и отчет о выполнении данного плана;

– инструкция администратора ИБ ООО «БИТ»;

– план действий в нештатных ситуациях;

– регламент доступа в серверные помещения;

– памятка/инструкция пользователя о правилах безопасной работы.

Данный список документации может быть расширен дополнительной документацией по требованию Заказчика, в соответствии с правилами оформления и содержания организационно-распорядительной документации на предприятии ООО «БИТ».

Заключение

В рамках обеспечения информационной безопасности предприятий оптовой торговли, таких как защита от инцидентов ИБ (хакерские атаки, НСД, некомпетентность или халатность сотрудников), снижение себестоимости проектов по обеспечению ИБ , в данной работе исследовались и разрабатывались эффективные способы обеспечения безопасности информации, передаваемой по каналам связи в автоматизированных системах управления.

Разработанная комплексная защита информационной безопасности предприятия в виде совокупного применения технического средств на базе Cisco и организационных мер в виде организационно-распорядительной документации, является актуальной, соответствует современным требованиям в области защиты информации, имеет практическую ценность и внедрена на предприятии оптовой торговли ООО «БИТ», так же использованные подходу могут применятся в системах различных отраслей при решении задач по обеспечению информационной безопасности в автоматизированных системах управления.

Литература

1. *Аль-Аммори А*. и др. Методы и средства защиты информации // The Scientific Heritage. 2020. № 51.

2. *Наталичев Р*. Эволюция и парадоксы нормативной базы обеспечения безопасности объектов критической информационной инфраструктуры // Безопасность Информационных Технологий. 2021. №3. https://bit.mephi.ru/index.php/bit/article/view/1359

3. *Данилина А.В.* Информационная безопасность как задача обеспечения экономической безопасности предприятия // Студенческий вестник: электронный научный журнал 2021. № 43(188). URL: https://studvestnik.ru/journal/stud/herald/

4. Приказ ФСТЭК России № 227 от 06.12.2017 «Об утверждении Порядка ведения реестра значимых объектов КИИ РФ».

5. Приказ ФСТЭК России № 229 от 11.12.2017 «Об утверждении формы акта проверки, составляемого по итогам проведения госконтроля в области обеспечения безопасности значимых объектов КИИ РФ».

6. Приказ ФСТЭК России № 235 от 21.12.2017 «Об утверждении Требований к созданию систем безопасности значимых объектов КИИ РФ и обеспечению их функционирования».

7. Приказ ФСТЭК России от 27.03.2019 N 64 "О внесении изменений в Требования к созданию систем безопасности значимых объектов критической информационной инфраструктуры Российской Федерации и обеспечению их функционирования, утвержденные приказом Федеральной службы по техническому и экспортному контролю от 21 декабря 2017 г. N 235"

8. Приказ ФСТЭК России № 236 от 22.12.2017 «Об утверждении формы направления сведений о результатах присвоения объекту КИИ одной из категорий значимости либо об отсутствии необходимости присвоения ему одной из таких категорий».

9. Приказ ФСТЭК России № 59 от 21.03.2019 «О внесении изменений в форму направления сведений о результатах присвоения объекту критической информационной инфраструктуры одной из категорий значимости либо об отсутствии необходимости присвоения ему одной из таких категорий, утвержденную приказом ФСТЭК № 236 от 22.12.2017».

10. Приказ ФСТЭК России № 239 от 25.12.2017 (ред. 09.08.2018) (ред. 26.03.2019) «Об утверждении Требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры РФ».

11. Федеральный закон № 193-ФЗ от 26.07.2017 «О внесении изменений в отдельные законодательные акты РФ в связи с принятием ФЗ «О безопасности критической информационной инфраструктуры РФ».

12. Методический документ ФСТЭК России от 05.02.2021 «Методика оценки угроз безопасности информации».

13. Банк данных угроз безопасности информации. [Электронный ресурс] - https://bdu.fstec.ru/

14. Проект указа Президента РФ «О мерах экономического характера по обеспечению технологической независимости и безопасности объектов критической информационной инфраструктуры».

15. Федеральный закон от 26.07.2017 № 187-ФЗ «О безопасности критической информационной инфраструктуры РФ».

16. Указ Президента Российской Федерации от 17.03.2008 № 351 «О мерах по обеспечению информационной безопасности Российской Федерации при использовании информационно-телекоммуникационных сетей международного информационного обмена».

17. Федеральный закон № 193-ФЗ от 26.07.2017 «О внесении изменений в отдельные законодательные акты РФ в связи с принятием ФЗ «О безопасности критической информационной инфраструктуры РФ».

18. Федеральный закон № 194-ФЗ от 26.07.2017 «О внесении изменений в УК РФ и УПК РФ в связи с принятием ФЗ «О безопасности критической информационной инфраструктуры РФ».

19. Постановление Правительства РФ от 08.02.2018 г. № 127 «Об утверждении Правил категорирования объектов критической информационной инфраструктуры Российской Федерации, а также перечня показателей критериев значимости объектов критической информационной инфраструктуры Российской Федерации и их значений».

20. Постановление Правительства РФ № 452 от 13.04.2019 «О внесении изменений в постановление Правительства № 127 от 08.02.2018».

21. Методические рекомендации по определению и категорированию объектов критической информационной инфраструктуры топливно-энергетического комплекса, Минэнерго России, 2019, 39 с.

22. Приказ ФСТЭК России № 138 от 09.08.2018 «О внесении изменений в Требования к обеспечению защиты информации в АСУ ТП и ТП на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды, утвержденные приказом ФСТЭК № 31, и в Требования по обеспечению безопасности значимых объектов критической информационной инфраструктуры РФ, утвержденные приказом ФСТЭК № 239».

23. Приказ ФСТЭК России № 60 от 26.03.2019 «О внесении изменений в Требования по обеспечению безопасности значимых объектов критической информационной инфраструктуры РФ, утвержденные приказом ФСТЭК № 239 от 25.12.2017».

24. Endpoint Security – Check Point Software. [Электронный ресурс]. https://www.checkpoint.com/solutions/endpoint-security/

25. Configure a Site-to-Site IPSec IKEv1 Tunnel Between an ASA and a Cisco IOS Router. [Электронный ресурс]. https://www.cisco.com/c/en/us/support/docs/security-vpn/ipsec-negotiation-ike-protocols/119425-configure-ipsec-00.html

GNSS ДЛЯ СИНХРОНИЗАЦИИ БАЗОВЫХ СТАНЦИЙ СЕТЕЙ 4G, 5G И В СРЕДСТВАХ ИЗМЕРЕНИЙ РАДИОСИГНАЛОВ

Исаева Людмила Николаевна,
Московский технический университет связи и информатики, заместитель начальника Испытательного центра, к.т.н., Москва, Россия
l.n.isaeva@mtuci.ru

Немыкин Андрей Александрович,
Московский технический университет связи и информатики, старший преподаватель кафедры Метрологии, стандартизации и измерений в инфокоммуникациях, Москва, Россия
a.a.nemykin@mtuci.ru

Лобзов Александр Валерьевич,
Московский технический университет связи и информатики, начальник Испытательного центра, к.т.н., Москва, Россия
a.v.lobzov@mtuci.ru

Коган Семен Самуилович,
ООО «Т8», советник генерального директора по формированию технической стратегии, к.т.н., Санкт-Петербург, Россия
kogan@t8.ru

Аннотация
В настоящей статье выявлены требования и рассмотрены способы синхронизации в сетях 4G & 5G. Обсуждены особенности использования GNSS (Global Navigation Satellite System) для синхронизации базовых станций сетей 4G & 5G. Исследованы алгоритмы фильтрации параметров, которые колеблются из-за неравномерного движения подвижного объекта, в измерительном электронном оборудовании. Показано, что использование сигнала GNSS в качестве источника опорной частоты повышает точность генератора опорной частоты средств измерений радиосигналов.

Ключевые слова: *Синхронизация, GNSS, 4G, 5G, измерения*

Введение

По мере эволюции сетей подвижной радиотелефонной связи к 5G менялись требования к синхронизации, которая гарантирует, что абонентские терминалы могут беспрепятственно подключаться к базовой станции и обеспечивать плавную передачу обслуживания при передвижении пользователя из одной соты в другую. В сетях 4G LTE и 5G NR с временным разделением каналов TDD (Time Division Duplex) необходимо обеспечить синхронизацию базовых станций как по частоте, так и по фазе/времени.

Синхронизация по фазе/времени на сетях 5G NR TDD должна:

– свести к минимуму количество защитных частотных полос для систем TDD;

– предотвратить интерференционные помехи внутри и между сотами сети подвижной связи с тем, чтобы пользователи могли переключаться между базовыми станциями;

– оптимизировать использование пропускной способности сети 5G.

Данная статья посвящена применению сигналов глобальной навигационной спутниковой системы (GNSS) для синхронизации базовых станций в сетях 4G & 5G. Рассмотрены вопросы повышения точности генератора опорной частоты средств измерений радиосигналов при использовании сигнала GNSS в качестве источника опорной частоты.

Требования к синхронизации базовых станций в сетях 4G & 5G

К синхронизации базовых станций в сетях 4G & 5G предъявляют следующие требования [1]:

– частота местных тактовых генераторов на сети сотовой связи должна быть такой же, как частота первичного эталонного генератора, расположенного, как правило, на уровне ядра сети (CORE);

– временной интервал между синхронизирующими импульсами должен сохраняться одинаковым по всей сетевой инфраструктуре;

– синхронизирующие импульсы/метки точного времени (один импульс в секунду) должны поступать на все радиоблоки в одно и то же время;

– для всех соседних сетей с функционалом TDD необходимо использовать аналогичную структуру цикла (кадра) с единой синхронизацией начала цикла (кадра);

– все базовые станции во избежание помех должны быть синхронизированы по фазе таким образом, чтобы ограничить сквозную погрешность (рассогласование) по времени при передаче сигналов от CORE до радиоблоков величиной 1,5 мкс, которая включает в себя две составляющие: 1,1 мкс – абсолютная ошибку по времени при передаче сигнала по сети подвижной связи от уровня CORE до уровня радиодоступа, и 0,4 мкс – на участке Fronthaul радиодоступа, то есть непосредственно перед радиоблоками;

– для реализации функциональности координированной передачи и приема CoMP (Coordinate MultiPoint) и множественного входа множественного выхода MIMO (Multiple-input-multiple-out) погрешность (рассогласование) по времени между радиоблоками (RRU), относящимися к одному кластеру подвижной связи (например, подключенными к одному и тому же электронному распределительному блоку (DU) в конфигурации централизованной (Centralized-RAN) или облачной (Claud-RAN) сети радиодоступа, не должна превышать ± 130 нс;

– точность тактовой частоты сигнала синхронизации на радиоинтерфейсе должна быть в пределах ± 50 ppb;

– точность фазы/времени сигнала синхронизации на радио интерфейсе должна быть [2, 3, 11] для:

- LTE (TDD): для соты более 3 км в пределах 10 мкс, для соты менее 3 км в пределах 3 мкс,
- 5G NR (TDD): в пределах 3 мкс,
- 5G NR MIMO: в пределах 65 нс.

Способы организация синхронизации в сетях 4G & 5G

В сетях 4G & 5G возможно несколько способов синхронизации базовых станций [4]. Сигналы синхронизации могут быть доставлены к базовым станциям:

– от первичного эталонного тактового генератора PRTC (Primary Reference Time Clock) или ведущего генератора T-GM (Transport GrandMaster), обычно располагаемых на уровне ядра сети (CORE), по оптоволоконной транспортной сети либо через полностью выделенную сеть;

– от ключевых централизованных точек в сети, где установлены приемники GNSS, по оптоволоконной транспортной сети;

– от приемников системы GNSS, такой как российская ГЛОбальная НАвигационная Спутниковая Система (ГЛОНАСС), или американская система глобального позиционирования (GPS), или европейская система (Galileo), или китайская навигационная спутниковая система (BeiDou), подключенных напрямую к базовым станциям.

В основе сквозной сети мобильной связи лежит транспортный уровень, организуемый поверх оптоволоконной среды передачи с использованием технологии многоканальной передачи с мультиплексированием и разделением оптических каналов по длине волны оптического излучения (WDM) [5]. Эволюция к сетям 5G связана не только с необходимостью повышения производительности, связности и гибкости узлов транспортной сети, но и с повышением пропускной способности оптических транспортных участков сети OTS (Optical Transport Section), имея в виду оптические каналы систем WDM и с обеспечением низкой задержки для сервисных соединений поверх транспортного уровня сети. Следует также учесть, что при переходе к Centralized-RAN и Cloud-RAN, оптоволоконные решения все шире используются на уровне радиодоступа.

Обеспечение высококачественной синхронизации по частоте, фазе и времени является обязательным условием для сетей 5G (МСЭ-Т G.8275). Существует три основных компонента сетевой синхронизации, которые могут быть реализованы в транспортной сети [6, 7]:

– синхронизация по частоте: частота тактовых генераторов (местных часов) в узлах сети должна быть такой же, как частота первичного генератора в PRTC, который обычно располагается на уровне CORE, причем временной интервал между синхронизирующими импульсами в узлах сети должен соответствовать первичному генератору в PRTC, но синхронизирующие сигналы не обязательно должны появляться во всех узлах сети одновременно; в соответствии с рекомендацией МСЭ-Т G.8272, в составе PRTC может быть включен модуль спутникового приемника сигнала – МСЭ-Т G.803/G.8260/G.8261/G.8263/G.8264/G.8265;

– синхронизация по фазе: синхронизирующие импульсы (1 PPS) появляются в узлах сети одновременно - МСЭ-Т G.8271/G.8272/G.8273/G.8274/ G.8275);

– синхронизация по времени суток: сообщения синхронизации включают точное время (ToD) – (МСЭ-Т G.8271/G.8272/G.8273/G.8274/ G.8275).

По мере того, как происходит эволюция от сетевой инфраструктуры, ориентированной в основном на технологии передачи с временным разделением каналов ВРК/TDM, к пакетно-ориентированной (преимущественно, Ethernet) инфраструктуре, меняются возможности и способы распределения сигналов синхронизации. Синхронизация по частоте в современных, пакетно-ориентированных, транспортных сетях реализуется с использованием функционала синхронного Ethernet (SynchE), обеспечивающего передачу тактовой частоты на физическом уровне клиентского сигнала Ethernet. Должно быть гарантировано, что тактовая частота на входных клиентских портах транспондеров или мукспондеров системы DWDM должна соответствовать тактовой частоте клиентского сигнала, передаваемого в полезной нагрузке структуры оптической транспортной сети (OTN) через линейные порты транспондеров или мукспондеров на сети DWDM.

Плохое качество синхронизации по параметрам фаза/время влияет на качество работы систем LTE-TDD, LTE-A и 5G NR. Синхронизация по фазе/времени на пакетно-ориентированной транспортной сети предусматривает доставку в пределах определенного интервала времени данных об абсолютной ошибке по времени как между центральным задающим генератором PRTC или T-GM и базовой станцией мобильной связи, так и об относительной ошибке по времени между соседними базовыми станциями. Алгоритм работы протокола точного времени 1588v2 (PTP) основан на обмене пакетами с отметками времени между ведущими (например, T-GM) и ведомыми (например, T-BC, Transport Boundary Clock) часами. Функционал T-BC располагается в узлах транспортной сети и включает встроенные часы PTP (Precision Time Protocol) клиента, соединенные с ведущим часами PTP. Такая конфигурация позволяет сетевому узлу синхронизировать местные часы (то есть T-BC в данном узле) от сигнала, приходящего от вышестоящих источников сигнала (например, T-GM/T-BC) и выступать в качестве основных (ведущих) часов по отношению к любым часам типа T-BC, находящимся в нижестоящих узлах сети.

Особенности использования GNSS для синхронизации базовых станций сетей 4G и 5G

GNSS помимо определения местоположения (географических координат) наземных, водных и воздушных объектов, а также низкоорбитальных космических аппаратов также позволяет получить значение скорости и направления движения приемника сигнала, сигналы точного времени.

В сетях подвижной радиотелефонной связи 3G и 4G спутниковые приемники встроены в узлы базовых станций NodeB и eNodeB. Контроллеры этих узлов принимают сообщения ToD, то есть получают каждую секунду импульс синхронизации (1 PPS) и используют его для синхронизации частот всех базовых станций подвижной связи. Затем контроллеры передают их далее по радиоканалу на оборудование пользователя. Сети 3G и 4G нуждаются в прямой связи только с одним спутником для синхронизации по частоте.

В сотовых сетях 5G используются те же спутники GNSS, что и в сетях 3G и 4G, однако немного по-другому. Для этого типа синхронизации требуется прямая видимость нескольких спутников. Чтобы правильно использовать сигнал ToD/PPS, получаемый от спутникового приемника, необходимо иметь возможность компенсировать задержку между моментом, когда спутник отправляет метку (сообщение) ToD/PPS, и моментом, когда данное сообщение поступает на спутниковый приемник.

Справиться с этой задачей непросто, поскольку спутники не находятся над сетью неподвижно. После расчета точного положения спутникового приемника можно определить задержку сигнала между спутниками и спутниковым приемником, чтобы "скорректировать" значение ToD, в котором сообщение было получено.

В расчетах необходимо учитывать четыре переменные – долготу, широту, высоту и время, а для такого расчета потребуются минимум четыре спутника. Чем больше времени отводится для опроса спутников, тем точнее определяется позиция спутника по отношению к спутниковому приемнику. Чем точнее установлено положение спутникового приемника, тем меньше будет ошибка по времени между сотами сети подвижной радиотелефонной связи и ниже вероятность того, что перекрывающиеся соты (ячейки) системы мобильной связи будут создавать взаимные помехи в результате интерференции.

Уязвимость системы GNSS связана со следующими обстоятельствами:

– использование радиочастотного интерфейса;

– вредоносные атаки, например, глушилки высокой мощности, спуфинг (подмена, при которой один человек или программа успешно маскируется под другую путем фальсификации данных);

– окружающая среда, например, трудности с установкой на сайтах базовых станций, проблемы с размещением антенн, солнечные лучи, повреждения от молний и т. п.

В последние годы участились случаи как преднамеренных, так и непреднамеренных взломов, глушения GNSS, что связано с использованием дешевых нелегальных глушителей GNSS, в связи с военными действиями. Возникающие новые обстоятельства вынуждают некоторые страны вводить законодательство, обеспечивающее защиту и надежность сетей синхронизации.

Для обеспечения синхронизации базовых станций по фазе/времени с применением GNSS необходимо иметь в виду, что фаза является таким параметром радиосигнала, который легче всего «разрушается» в процессе распространения радиоволны и при прохождении радиосигнала по цепям радиоэлектронной аппаратуры. Калибровка высокоточных фазоизмерительных устройств с целью компенсации задержки сигнала в приемном тракте, которая обусловлена влиянием фазо-частотной характеристики частотно-избирательной цепи (неточно настроенной), как это делается, например, в приемном устройстве аппаратуры GNSS с частотным разделением каналов, в условиях действия интенсивных помех может оказаться неэффективной, поскольку не учитывает смещения оценки фазы, обусловленного асимметрией спектра помехи при неточной настройке частотно-избирательной цепи [8, 9].

Приемники GNSS можно защитить от некоторых указанных помех, но принимаемые для этого меры увеличивают затраты на содержание сети. Кроме того, операторы подвижной радиотелефонной связи должны учитывать, что при переходе к сетям 5G быстро увеличится количество сот системы мобильной связи, в том числе в тех местах, где использование спутниковых приемников GNSS затруднительно. Например, в условиях плотной городской застройки для развития сетей 5G и предоставления услуг широкополосной связи на более коротком расстоянии потребуются небольшие соты с использованием радиоспектра миллиметрового диапазона. Такие соты могут быть развернуты в труднодоступных местах, например, в глубине торговых центров, на разных этажах многоквартирных домов и т. п.

Повышение точности генератора опорной частоты средств измерений радиосигналов

Современные средства измерений радиосигналов, такие как анализаторы спектра, комплектуются встроенной функцией приемника сигналов GNSS (ГЛОНАСС, GPS, Galileo и Beidou), которая помимо получения информации о широте, долготе, высоте и универсальном глобальном времени (UTC), повышает точность генератора опорной частоты. Это относится, например, к анализаторам спектра MS2090A, производства Anritsu Company (США), H600 RFHawk производства компании Tektronix (США), R&S®FSW производства компании Ronde & Schavrz (США), ThinkRS5700 производства корпорации Think RF (Канада) и т.п.

Рассмотрим работу встроенной функции приемника сигналов GNSS на примере анализатора спектра MS2090A производства Anritsu Company (США) [10]. После настройки функции GNSS начинается вывод следующей информации с постоянным обновлением: состояние подключения, отслеживаемые спутники, широта, долгота, высота, UTC. После установки местоположения GNSS внутренний генератор опорной частоты запускает процедуру корректировки своей частоты в

соответствии с эталонным сигналом GNSS. Не более чем через 3 минуты после нахождения спутников точность генератора опорной частоты будет составлять не хуже, чем ± ($2{,}5 \cdot 10^{-8}$). Собственная стандартная точность термостатированного кварцевого генератора составляет ± ($3 \cdot 10^{-7}$). Поправочный коэффициент, применяемый к внутреннему кварцевому генератору, позволяет прибору поддерживать точность частоты по GNSS на уровне, по крайней мере, ± ($5 \cdot 10^{-8}$) в течение трех дней (режим holdover), даже если прибор не имеет возможности получать сигналы от спутников GNSS.

При синхронизации от внутреннего источника или внешнего источника (не GNSS, например генератор MG3694C) обеспечивается точность генератора опорной частоты

$$\pm (\delta_0 + 0{,}1 \cdot N \cdot \delta_A) \quad (1)$$

где $\delta_0 = \pm (3 \cdot 10^{-7})$ – допускаемая относительная погрешность частоты опорного генератора в диапазоне температур от 0 до 50ºС при выпуске из производства или после подстройки; $\delta_A = \pm (1 \cdot 10^{-6})$ – относительный временной дрейф частоты опорного генератора за 10 лет; N – количество лет со дня подстройки.

Таким образом применение сигналов GNSS для синхронизации анализаторов спектра радиосигнала повышает точность синхронизации на порядок по сравнению с синхронизацией от внутреннего источника или внешнего источника (не GNSS).

Также анализаторы спектра могут использовать тактовый сигнал GNSS в качестве запускающего сигнала для активизации функции ждущей развертки, позволяющей синхронизовать развертку с событием так, чтобы анализатор осуществлял сбор данных в нужное время. Эта функция обычно полезна для измерения сигналов во временной области, таких как импульсные ВЧ сигналы, сигналы с временным мультиплексированием или импульсные модулированные сигналы.

Заключение

В настоящей статье выявлены требования к синхронизации базовых станций в сетях 4G & 5G как по частоте, так и по фазе/времени. Обеспечение высококачественной синхронизации по частоте, фазе и времени является обязательным условием для сетей 5G. Рассмотрены способы организация синхронизации в сетях 4G & 5G: передача синхросигнала по оптоволоконной транспортной сети и/или от приемников GNSS. Выделены достоинства и недостатки использования GNSS для синхронизации базовых станций сетей 4G & 5G. На примере анализатора спектра MS2090A производства Anritsu Company (США) показано, что использование сигнала GNSS в качестве источника опорной частоты повышает точность генератора опорной частоты средств измерений радиосигналов.

Литература

1. *Коган С.С.* СЕТИ 5G: распределение сигналов синхронизации на оптическом транспортном уровне. Часть 1. Общие требования по синхронизации для сетей мобильной связи 5G // Первая миля/Last Mile. 2022. № 4. С. 50-59, doi: 10.22184/2070-8963.2022.104.4.50.59.

2. 3GPP Technical specification (TS) #: 38.133 “NR; Requirements for support of radio resource management”.

3. 3GPP Technical specification (TS) #: 38.104 “NR; Base Station (BS) radio transmission and reception”.

4. Synchronization Distribution in 5G Transport Networks. E-book // Infinera. 2021. URL: https://www.infinera.com/wp-content/uploads/Synchronization-Distribution-in-5G-Transport-Networks-0282-EB-RevA-0321.pdf (дата обращения: 03.07.2022).

5. Isaeva L.N., Lobzov A.V., Kogan S.S. Analysis of Critical Parameters of Promising Optical Interfaces of High-Speed Telecommunication Systems // 2023 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), St. Petersburg, Russian Federation, 2023, pp. 1-6, doi: 10.1109/WECONF57201.2023.10147990

6. *Коган С.С.* Сети 5G: распределение сигналов синхронизации на сетевом оптическом транспортном уровне. Часть 2. Сетевая синхронизация по тактовой частоте // Первая миля/Last Mile. 2022. № 5. С. 44-58, doi: 10.22184/2070-8963.2022.105.5.44.58.

7. *Коган С.С.* Сети 5G: распределение сигналов синхронизации на сетевом оптическом транспортном уровне. Часть 3. Сетевая синхронизация по фазе/времени // Первая миля/Last Mile. 2022. № 6. С. 42-53, doi: 10.22184/2070-8963.2022.106.6.42.53

8. *Nemykin A.A.* Comparative Analysis of the Accuracy And Dynamic Characteristics of Navigate Radio Electronic Equipment with Phase and Frequency Auto Surveying in Intensive Inferences // 2019 Systems of Signals Generating and

Processing in the Field of on Board Communications, Moscow, Russia, 2019, pp. 1-4, doi: 10.1109/SOSG.2019.8706787.

9. *Nemykin A.A., Stroganova E.P.* Analysis of Moving Radio Electronic Measuring Instruments Characteristics // 2020 International Conference on Engineering Management of Communication and Technology (EMCTECH), Vienna, Austria, 2020, pp. 1-4, doi: 10.1109/EMCTECH49634.2020.9261508.

10. MS2090A Spectrum Analyzer. Operation Manual.

11. Dymkova S. Earth observation and global navigation satellite systems analitical report part II (timing & synchronisation of telecommunication networks, maritime and inland waterways, rail and automotive transport) // Synchroinfo Journal, vol. 8, no 2, pp. 24-34, 2022. DOI: 10.36724/2664-066X-2022-8-2-24-34.

ФОРКАМЕРНАЯ СИСТЕМА ВПРЫСКА ТОПЛИВА В ДВИГАТЕЛЯХ ВНУТРЕННЕГО СГОРАНИЯ

Матвей Прохорович Шарыгин,
Московский автомобильно-дорожный государственный технический университет (МАДИ), Москва, Россия
pro18061973@gmail.com

Аннотация
В статье рассматривается конструкция двигателя внутреннего сгорания, в основе которого имеется форкамерная система впрыска топлива. Описание данной технологии включает в себя устройство как её первого принципа, так и модификации различного рода форкамерных моторов некоторых автомобильных концернов. В нынешнее время данная система применяется в дизельных и бензиновых силовых агрегатах, а также имеет возможность применения в современных мотоциклетных и спортивных автомобильных двигателях. Эта технология применяется для повышения эффективности воспламенения топлива в камерах ДВС путём снижения степени сжатия за счёт более равномерного перемешивания горючей смеси, тем самым, увеличивая топливную экономичность и максимальную мощность силового агрегата, уменьшая детонации при воспламенении смеси и понижая количество токсичных выбросов в окружающую среду.

Ключевые слова: *форкамерный двигатель, форкамерно-факельное зажигание, послойное сгорание топлива, увеличение эффективности сгорания топлива, форкамера*

Введение

Вопрос экономии топлива всегда стоял остро перед автомобилистами, так как именно затраты на его покупку составляют чуть ли не 75% от всех расходов на содержание автомобилем [1, 11-24]. Многие автопроизводители прибегают к значительному модифицированию имеющихся силовых агрегатов, чтобы понизить расход горючего без потери мощностных характеристик двигателя внутреннего сгорания. Одним из направлений решения этой задачи является изобретение форкамерного двигателя.

Форкамерный двигатель – тип ДВС, отличающегося применением форкамеры, которая представляет собой дополнительную камеру сгорания, размещенную над основной камерой [2]. В ее задачи входит предварительное смешивание и сжигание топлива с воздухом перед тем, как окончательная смесь будет воспламеняться в цилиндрах.

В устройстве форкамерного двигателя (рис. 1) заложен гениальный принцип работы: топливовоздушная смесь поступает через дополнительный клапан в предварительную камеру сгорания.

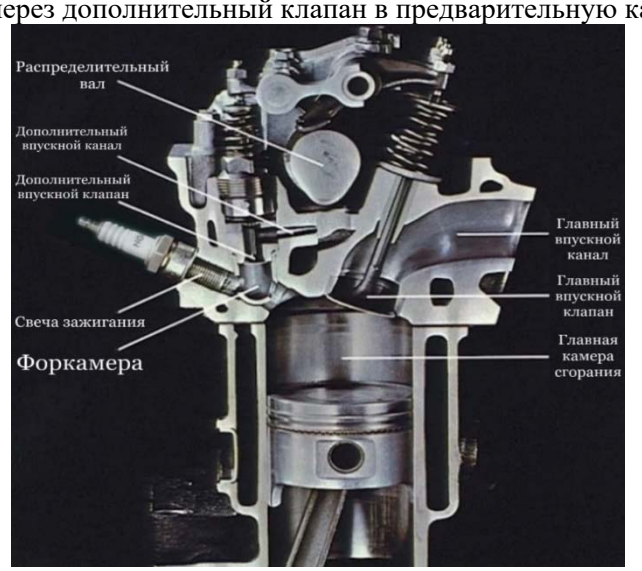


Рис. 1. Форкамерный двигатель Honda CVCC в разрезе

В ней смесь, в отличии от той, что в основном цилиндре, обогащена. Далее при помощи свечи зажигания она воспламеняется и струёй горящего топлива под высоким давлением попадает в основной цилиндр, воспламеняя остальную бедную смесь [3]. Таким образом не возникает детонаций и топливо наиболее равномерно воспламеняется в камере сгорания, нежели в обычном, привычном для нас ДВС без форкамеры.

История создания

Внедрение форкамеры в двигатель внутреннего сгорания произошло ещё в начале XX века. 14 марта 1909 года немецкий инженер и изобретатель Проспер Л'Оранж получил патент на свой принцип форкамеры (DRP 230 517).

Немецкий инженер и промышленник Карл Бенц, привлекает Л'Оранжа с его патентом на форкамерный двигатель, чтобы внедрить его в различные транспортные средства, ведь на то время дизельные двигатели использовались в основном в стационарном виде [4].

Благодаря дизельному агрегату с форкамерой удалось достичь гораздо более высоких оборотов двигателя и, следовательно, высокой производительности, чем при использовании старых версий двигателя с воспламенением от сжатия.

Система форкамерного сгорания топлива Л'Оранжа привела к созданию первого сельскохозяйственного транспортного средства, имеющего дизельный двигатель. Двухцилиндровые четырехтактные форкамерные двигатели мощностью 25 л.с. / 18 кВт при 800 об/мин установили на трактора Benz-Sendling S6. В 1923 году компания Daimler-Motoren-Gesellschaft представила грузовик с форкамерным двигателем OB 2, который является первым серийным дизельным двигателем для коммерческих автомобилей.

Данный силовой агрегат имел несколько модификаций и обеспечивал максимальную мощность до 40 л.с. Номинальная частота вращения составляла 1000 об /мин, и на тот момент времени, этот мотор считался весьма оборотистым.

Этот революционный силовой агрегат отличался повышенной экономичностью и Мангеймскомое отделение Ассоциации немецких инженеров заявило: «Дорожные испытания с пятью полностью загруженными пятитонными грузовиками на протяжении 103 километров сложной холмистой местности в абсолютно идентичных условиях показали общий расход 40,66 кг для автомобиля, работающего на бензине, и 29,95 кг для дизельного грузовика, т.е. экономию 32% веса топлива и 86% стоимости топлива» [5].

Сенсационно низкие затраты на топливо были обусловлены тем, что дизель был значительно дешевле бензина. Рекламируя свой экономичный двигатель в 1923 году компания утверждала: *«новый двигатель может также работать на газовом топливе, керосине, техасской нефти и желтом или коричневом парафиновом масле»*

В результате слияния компаний Benz, Cie и DMG в 1926 году принцип форкамеры Benz взял верх над дизельным двигателем с впрыском воздуха. При совместной разработке появился первый дизельный двигатель с форкамерой, применяемый на автомобилях. Им являлся шестицилиндровый силовой агрегат OM 5 1927 года выпуска с мощностью 75 л.с. и рабочим объемом 8,6 литра.

Применение форкамерного двигателя в легковых автомобилях

После внедрения дизельного форкамерного двигателя в коммерческую технику, компания Daimler-Benz осенью 1933-го года решила экспериментировать с дизелями для легковых автомобилей. Таким образом, Mercedes-Benz 260 D W138, представленный в 1936 году стал первым серийным дизельным легковым автомобилем. Его четырехцилиндровый силовой агрегат OM-138 (рис. 2) был всё так же форкамерным, засчёт чего в то время являлся весьма мощным и высокооборотистым (мощность в 45 л.с. при 3200 об/мин), а также имел небольшой расход топлива (9,5 л/100км).

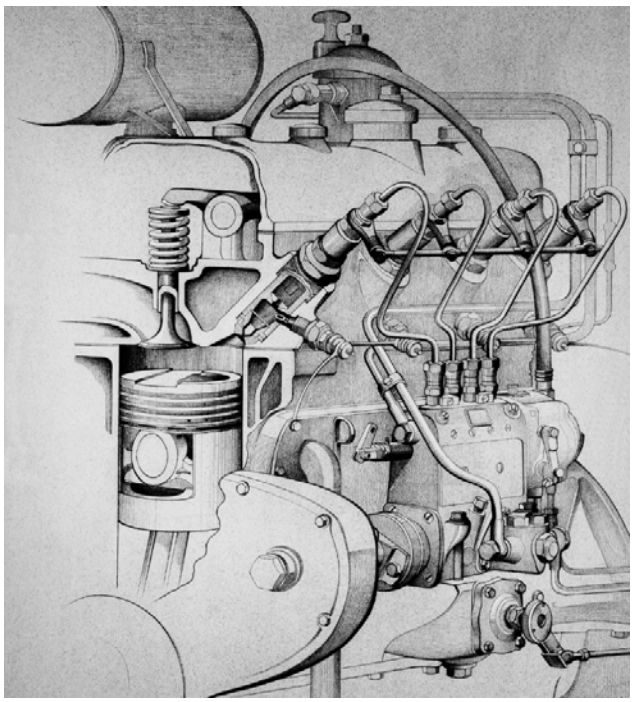


Рис. 2. Дизельный форкамерный двигатель Mercedes-Benz OM-138

С течением лет технология форкамерного впрыска дизельного топлива активно модифицировалась и применялась в двигателях Mercedes-Benz вплоть до 2001 года и последними моторами с данной конструкцией стали силовые агрегаты одной серии om604/605/606, применяемые в легковых автомобилях и отличающиеся только количеством цилиндров 4, 5 и 6 соответственно.

Форкамерный двигатель в СССР

В начале 1950-х годов линейка моторов автомобильных заводов СССР весьма устарела на фоне силовых агрегатов, производимых за границей. Это побудило советских инженеров Горьковского автомобильного завода изобрести подобный бензиновый форкамерный силовой агрегат.

Было принято решение усовершенствовать имеющийся нижнеклапанный двигатель ГАЗ-11, который устанавливался на грузовик ГАЗ-51, широко использующиеся по всей территории СССР. Этот мотор был прилично устаревшим, ведь являлся копией западного двигателя «Chrysler flathead», выпускавшийся Chrysler Corporation с 1924 года.

Проектированием нового форкамерно-факельного двигателя, получившим название ГАЗ-51Ф, занималось НАМИ под руководством конструктора Нилова. Получившиеся в 1956 году двигатель имел ГБЦ с тремя клапанами на цилиндр, приводимыми толкателями, и новый двухкамерный карбюратор.

Схема камеры сгорания двигателя с факельным зажиганием смеси показана на рис. 3 [6]. Камера сгорания состоит из основной камеры 1 и предкамеры 8. Горючая смесь в камеры поступает из карбюратора 4 по трубопроводам 3 и 5, причем в основную камеру поступает бедная смесь, а в предкамеру - обогащенная. Наполнение камер бедной и обогащенной смесями происходит во время впуска через клапаны 2 и 6. В конце такта сжатия между электродами свечи зажигания 7 проскакивает искра, воспламеняющая обогащенную смесь в предкамере. Давление в предкамере резко возрастает, в результате чего из предкамеры через каналы А выбрасываются в основную камеру факелы пламени через два отверстия. Из-за большой поверхности контакта и интенсивного вихревого движения факелы воспламеняют бедную смесь в основной камере и обеспечивают сгорание ее с высокими скоростями.

На рисунке 4 показана головка одного из цилиндров двигателя ГАЗ-51Ф с факельным зажиганием. Головка цилиндров имеет предкамеры 2, которые соединены с основными камерами сгорания 6 двумя каналами А, расположенными под углом. Горючая смесь поступает в камеры через клапаны 3 и 5, которые открываются одновременно коромыслами 4. В каждой предкамере установлена свеча зажигания 1.

В результате сравнения удельных эффективных расходов топлива исследования показали, что ГАЗ-51 со стандартным нижнеклапанным двигателем ГАЗ-11 значительно уступает по экономичности топлива такому же грузовому автомобилю с установленным в нем новым форкамерно-факельном силовым агрегатом ГАЗ-51Ф.

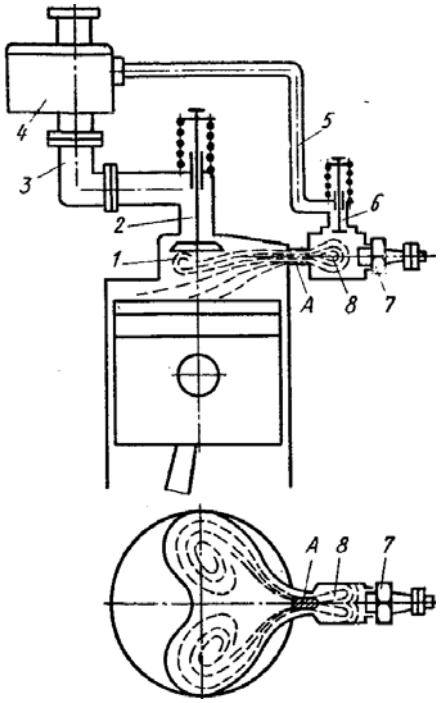


Рис. 3. Камера сгорания двигателя с форкамерно-факельным зажиганием:
А - канал; 1 – основная камера; 2 и 6 – впускные клапаны; 3 и 5 - трубопроводы; 4 – карбюратор; 7 – свеча зажигания; 8 – предкамера.

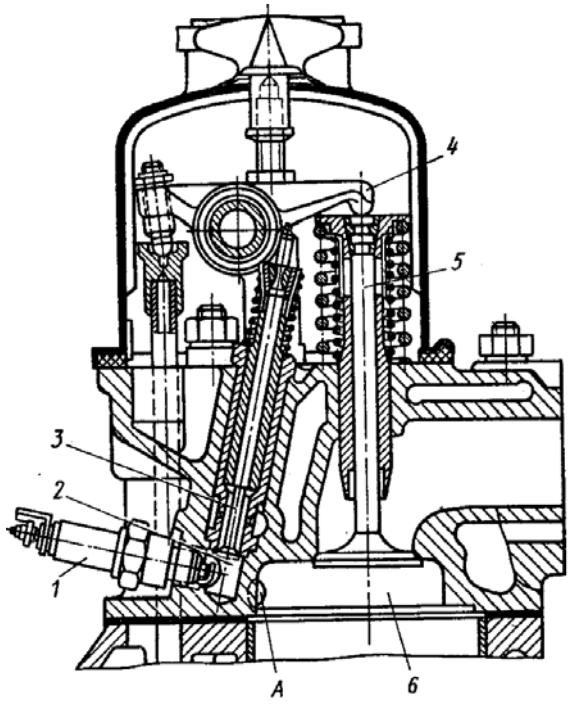


Рис. 4. Головка цилиндра ГАЗ-51Ф с форкамерно-факельным зажиганием:
А - канал; 1 – свеча зажигания; 2 — предкамера; 3 и 5 – клапаны; 4 – коромысло клапанов; 6 – основная камера

Грузовик ГАЗ-51Ф, получивший данный двигатель, активно эксплуатировался в экспериментальных целях по дорогам Москвы и Сочи. Но помимо весомого достоинства в виде пониженного расхода топлива, система имела и некоторые недостатки: образование сажи в форкамере, проблемы холодного пуска и сложность настройки карбюратора, которая требовала обслуживания исключительно квалифицированных специалистов. Именно из-за своей непрактичности и малой ремонтопригодности данный агрегат обрёл не самую лучшую славу у водителей, и после мелкосерийного выпуска грузовиков с данными ДВС, проект на время заморозили [7,8].

Форкамерный двигатель Honda и забота об экологии

В связи с увеличением мирового автопарка власти многих государств уже на протяжении многих десятков лет обеспокоены увеличением вредных выбросов от выхлопа автомобилей в окружающую среду. Поэтому существующие во многих странах экологические организации, на протяжении многих лет выдвигают новые нормы токсичности выхлопных газов, диктующие автопроизводителям современные условия выпуска и разработки новых автомобилей.

В 1971 г. японский производитель Honda, в результате собственных экологических исследований, создал новую систему сгорания топлива CVCC (Compound Vortex Controlled Combustion), что расшифровывается как составное вихревое управляемое сгорание. Силовой агрегат с такой системой (рис. 1) являлся уникальным в том смысле, что компания впервые применила так называемый принцип послойного сгорания в бензиновом ДВС, при помощи введения в него дополнительной камеры сгорания.

Двигатель с данной технологией первым среди других двигателей удовлетворял новым нормам 1975 года на токсичность выхлопных газов даже без установки каталитического нейтрализатора. Также двигатель отличался от классических бензиновых безфоркамерных ДВС крайней экономичностью, ведь бушующий в 1973 году мировой топливный кризис, побуждал автовладельцев делать свой выбор в пользу экономичных автомобилей.

Возрождение форкамерного двигателя в СССР

Успех «Хонды» вдохновил в 1972 году советских инженеров Горьковского автомобильного завода на развитие в сфере конструирования нового форкамерно-факельного силового агрегата. На тот момент конструкторское бюро активно занималось созданием новой «Волги», которая должна была прийти на смену ГАЗ-24. Было принято решение создать новый двигатель для нее на базе имеющегося популярного в стране мотора ЗМЗ-24.

Новый форкамерно-факельный двигатель, получивший индекс ЗМЗ-4022.10 (рис. 5) унаследовал конструктивные особенности своего грузового форкамерного предшественника ГАЗ-51Ф (рис. 4). Он был всё так же нижневальный, имел по три клапана на цилиндр и форкамера с основной камерой сгорания так же соединялась двумя отверстиями диаметром 3,5 мм.

Новый двигатель ЗМЗ-4022.10 был сконструирован на основе своего предшественника ЗМЗ-24Д и отличался от него новым карбюратором К-156, улучшенной системой охлаждения и зажигания, иной ГБЦ с увеличенным ходом клапанов и дополнительными каналами и клапанами для форкамер, а также другими доработками, актуальными на то время.

Улучшения пошли на пользу и новый форкамерно-факельный мотор стал мощнее, развивая 105 л.с. / 69,9 кВт, против 95 л.с. / 65 кВт у старого ЗМЗ-24Д, при одинаковом рабочем объёме 2445 куб.см и идентичной частоте вращения 4500 об/мин у обоих моторов. Также ЗМЗ-4022.10 отличался повышенной экономичностью (8,5 л на 100 км) и сниженным количеством вредных выбросов от выхлопа, в сравнении со старым силовым агрегатом.

В 1981 году стартовало серийное производство форкамерно-факельных моторов на Заволжском моторном заводе. Двигатели ЗМЗ-4022.10 повсеместно устанавливались в новоиспеченные «Волги», вышедшие в этом же году под индексом ГАЗ-3102. Мотор продержался на производстве вплоть до 1992 года, и ввиду морального устаревания нижневальной конструкции, заменен более актуальным на то время двигателем ЗМЗ-406 с верхним расположением распределительных валов и не имеющий форкамер. За 11 лет произведено около 27 тыс. ГАЗ-3102 с данным двигателем.

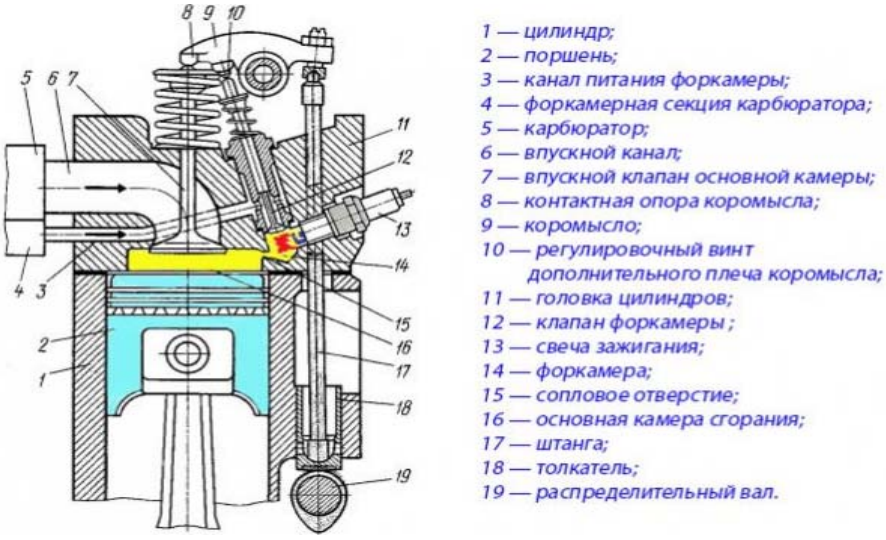


Рис 5. Схема форкамерно-факельного двигателя ЗМЗ-4022.10

Форкамерные двигатели в наше время

С внедрением большого количества электроники в конструкцию ДВС удалось изменить систему впрыска на непосредственную и значительно увеличить эффективность сгорания топлива [9]. Многие производители в итоге отказались от применения форкамерного впрыска, однако по сегодняшний день данная конструкция применяется в большом количестве дизельных двигателей, в основном, устанавливаемых в спецтехнику.

Также, по информации на 2021 год, Ford выделил 10 миллионов долларов на изучение возможности применения форкамерно-факельного зажигания на моторе семейства EcoBoost: V6 объёмом 3.5 литра. В теории внедрение новой технологии должно уменьшить расход топлива на 23% и снизить вес двигателя на 15% [10]. Изучение новой для концерна технологии займет порядка трёх лет.

Форкамерный двигатель в автоспорте

В гонках Формулы 1, где счёт идет на снижение каждого грамма массы болида, инженеры стараются повысить топливную эффективность двигателей, чтобы путем снижения расхода топлива, сократить его количество, заправляемое в бак, соответственно снизить снаряженную массу болида. Тем более с 2010 года новый регламент запретил дозаправку болидов во время гонки.

В 2010 году немецкая компания MAHLE, занимающаяся производством двигателей внутреннего сгорания, представила новую технологию Mahle Jet Ignition (рис.6), заменяющую стандартную свечу зажигания в двигателях на форкамеру струйного зажигания в сборе, не требуя отдельного воспламенителя в основной камере сгорания. В 2015 году компания адаптировала систему под требования Формулы 1, что позволило Ferrari впервые применить его на Гран При Канады.

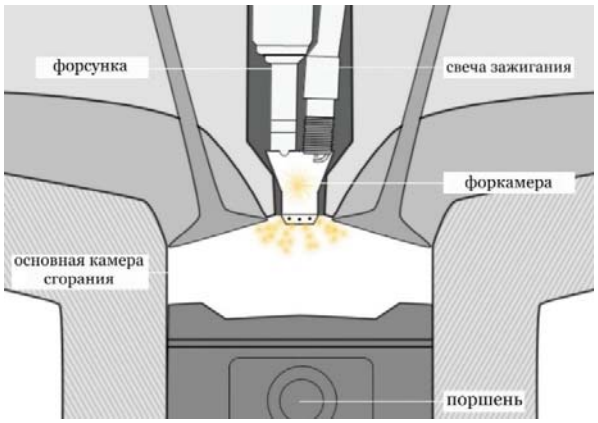


Рис. 6. Схема форкамерного двигателя с технологией Mahle Jet Ignition

Компания видимо посчитала, что новое – это хорошо забытое старое и сохраняя принцип работы форкамерного бензинового двигателя, переработала его конструкцию при помощи современных технологий. За подачу топлива в предварительную камеру отвечает не клапан, как это было ранее, а топливная форсунка. Нижняя часть предварительной камеры изолирована от основной и имеет направленные отверстия, подобные по своей конструкции тем, которые применяются в обыкновенных топливных форсунках, для равномерного распределения обогащенной смеси в основную камеру, в которой находится обедненная смесь.

Форкамерный двигатель в мототехнике

В 2020 году компания Honda создала новую систему форкамерного зажигания для двухтактного двигателя (рис. 7). В её конструкции используется две форсунки: первая совершенно обычная, подаёт топливо во впускной тракт прямо под дроссельной заслонкой, далее оно попадает через впускной клапан как на любом современном бензиновом двигателе. В свою очередь, вторая форсунка располагается в небольшой предварительной камере (форкамере) над основной камерой сгорания. Там же установлена и свеча зажигания, а между камерами находится вращающаяся заслонка (рис. 6), приводимая цепью ГРМ, отделяющая предкамеру от основной камеры.

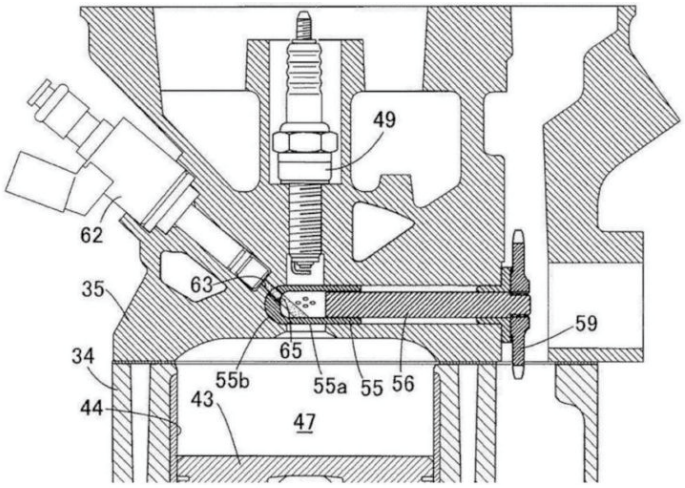


Рис. 7. Схема форкамерного мотоциклетного двигателя Honda

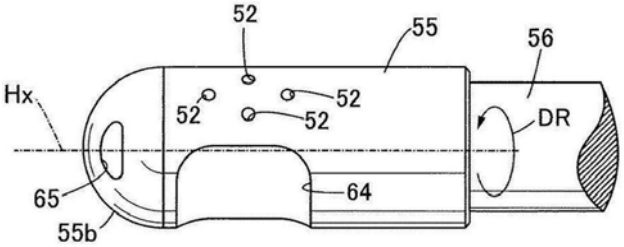


Рис. 8. Схема вращающиеся заслонки в форкамерном мотоциклетном двигателе Honda

После воспламенения горящей смеси поршень уходит вниз, а вращающаяся предварительная камера своей полностью открытой частью 64 находится снизу. Далее поршень поднимается и в конце сжатия происходит впрыск в предварительную камеру через отверстие 65 при помощи форсунки 62. В это время вращающаяся предварительная камера повернулась на 180 градусов частью с несколькими отверстиями 52 вниз, а при помощи открытой части 64 сверху происходит воспламенение богатой смеси при помощи свечи зажигания 49. Далее происходит взрыв внутри форкамеры 55a, и таким образом факелы пламени, исходящие из отверстий 52, могут распространяться и воспламенять обедненную смесь, находящуюся внутри основной камеры сгорания.

Заключение

Внедрение в двигатель внутреннего сгорания форкамерной системы впрыска топлива заметно увеличивает топливную эффективность, путем более равномерного сгорания горючей смеси внутри основных цилиндров. Факел, образуемый путём сгорания обогащенной смеси в предкамере, создаёт наилучшие условия для эффективного зажигания обедненной смеси в основной камере сгорания.

Таким образом, форкамерный мотор, за счёт повышенной топливной эффективности отличается сниженным расходом топлива, уменьшенным количеством вредных выбросов в окружающую среду, а также отмечается увеличение мощности.

Вероятно, автопроизводители, в угоду экологическим требованиям, с целью повышения вышеперечисленных характеристик силового агрегата, ещё не раз вернутся к вопросу внедрения в серийные автомобили данной системы сгорания топлива, переработав старую технологию под современный непосредственный впрыск.

Литература

1. *Луканин В.Н., Алексеев И.В., Шатров М.Г. и др.* Двигатели внутреннего сгорания: учебник для вузов: В 3 кн.; под ред. В.Н. Луканина. М.: Высшая школа, 1995.

2. *Кузнецов А.Г., Харитонов С.В., Рыжов В.А.* Разработка и исследование системы управления дизельным двигателем // Двигателестроение. 2021. № 2. С. 20-25.

3. *Говорушенко Н.Я.* Диагностика технического состояния автомобилей: учеб. для вузов. М.: Транспорт, 1970. 252 с.

4. *Мальчук В.И.* Топливоподача и зональное смесеобразование в дизелях. М.: МАДИ, 2009. 176 с.

5. *Алексеев И.В.* и др. Двигатели автотракторной техники: учебник / И.В. Алексеев, К.А. Морозов, Ю.В. Горшков, С.А. Пришвин, В.В. Синявский, А.Ю. Дунин, А.Л. Яковенко, С.Д. Скороделов, М.Г. Шатров; под. ред. М.Г. Шатрова. М.: КНОРУС, 2016. 400 с.

6. *Шароглазов Б.А., Поваляев В.А.* Расчетная оценка качества наполнения свежим зарядом цилиндров поршневого двигателя на стадии проектирования // Вестник нац. исслед. Южно-Уральского гос. ун-та. 2008. No 23. С. 20-24.

7. *Карелина М.Ю., Ершов В.С., Акулов А.А., Талдыкин Д.С*. Испытание дизельного двигателя при отрицательных температурах с добавлением модификатора Фтор-ПАВ в масло // Транспортные системы и дорожная инфраструктура Крайнего Севера: Сборник материалов III всероссийского форума, Якутск, 29 марта – 01 апреля 2022 года / Редколлегия: Д.В. Филиппов, В.Ю. Панков, Г.О. Николаева. Якутск: Северо-Восточный федеральный университет имени М.К. Аммосова, 2022. С. 139-144. EDN KQRGNN.

8. *Yefimenko D.B., Ptitsyn D.A., Akulov A.A., Smirnov P.I.* Modeling of Fuel Consumption of Passenger Cars Based on Their Technical Characteristics // 2021 Systems of Signals Generating and Processing in the Field of on Board Communications, Conference Proceedings, Moscow, 16-18 марта 2021 г. Moscow, 2021. P. 9416138. DOI 10.1109/IEEECONF51389.2021.9416138. EDN KPFCMV.

9. *Карелина М.Ю., Ершов В.С., Акулов А.А., Ерпулев В.А*. Анализ работы карбюраторного двигателя в условиях низких температур с добавлением высокоэффективного модификатора Фтор-ПАВ в масло // Транспортные системы и дорожная инфраструктура Крайнего Севера: Сборник материалов III всероссийского форума, Якутск, 29 марта – 01 апреля 2022 года / Редколлегия: Д.В. Филиппов, В.Ю. Панков, Г.О. Николаева. Якутск: Северо-Восточный федеральный университет имени М.К. Аммосова, 2022. С. 134-139. EDN BDPGXY.

10. *Osborne R.J., Stokes J., Lake T.H.* et al. Development of a Two-Stroke/Four-Stroke Switching Gasoline Engine – The 2/4 SIGHT Concept // SAE Technical Paper Series. 2005. № 2005-01-1137. P. 1-17.

11. Михайлов О.В. Основы мировой конкурентоспособности // Сер. Интеллектуальное богатство России. Москва, 1999.

12. *Ивантер В.В., Кузык Б.Н*. Будущее России: инерционное развитие или инновационный прорыв? // Монография. Москва, 2005.

13. *Кибанов А.Я., Дмитриева Ю.А*. Управление персоналом // конкурентоспособность выпускников вузов на рынке труда. Москва, 2011.

14. *Карелина М.Ю., Арифуллин И.В., Терентьев А.В*. Аналитическое определение весовых коэффициентов при многокритериальной оценке эффективности автотранспортных средств // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2018. № 1 (52). С. 3-9.

15. *Долина О.Н., Жидкова М.А., Шпилькина Т.А., Ахметжанова Э.У*. Реализация политики импортозамещения в автомобильной промышленности // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2017. № 2 (49). С. 22-28.

16. *Пузаков А.В., Осаулко Я.Ю.* Исследование влияния эксплуатационных факторов на тепловое состояние автомобильного генератора // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2018. № 1 (52). С. 16-23.

17. *Надараиа Ц.Г., Селиванов А.И., Шестаков И.Я., Фадеев А.А., Бабкина Л.А*. Химико-кинетический накопитель энергии и мотор-редуктор для электромобиля // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2017. № 1 (48). С. 12-17.

18. *Мельникова Т.Е., Мельников С.Е., Завязкина В.В.* Электромобили: перспективы и пути развития // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2019. № 3 (58). С. 22-26.

19. *Блудян Н.О.* Перспективы развития электрических автобусов // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2020. № 3 (62). С. 19-24.

20. *Ухов И.В., Климов А.В., Долгий И.О., Рябцев Ф.А*. Анализ и моделирование алгоритма i2t лимитирования тока для литий-ионных батарей // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2021. № 1 (64). С. 3-10.

21. *Климов А.В., Анисимов В.Р.* О некоторых аспектах повышения энергонасыщенности тяговых электрических двигателей // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2021. № 2 (65). С. 26-31.

22. *Пузаков А.В., Султанов Н.З.* Аналитическая модель обмотки ротора автомобильного генератора // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2022. № 4 (71). С. 11-17.

23. *Богумил В.Н., Элдиба Х.М.М.М.* Разработка архитектуры бортового навигационно-связного блока для городского пассажирского транспорта, включающего функцию контроля режимов труда и отдыха водителей // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2022. № 4 (71). С. 96-101.

24. *Ухов И.В., Анисимов В.Р., Рябцев Ф.А., Спинов А.Р*. Исследование эквивалентного цикла движения автомобиля с тяговым электрическим приводом на основании анализа длины маршрутов и пассажиропотока // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2022. № 3 (70). С. 3-7.